JP04/15333

12.10.2004

日本国特許庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

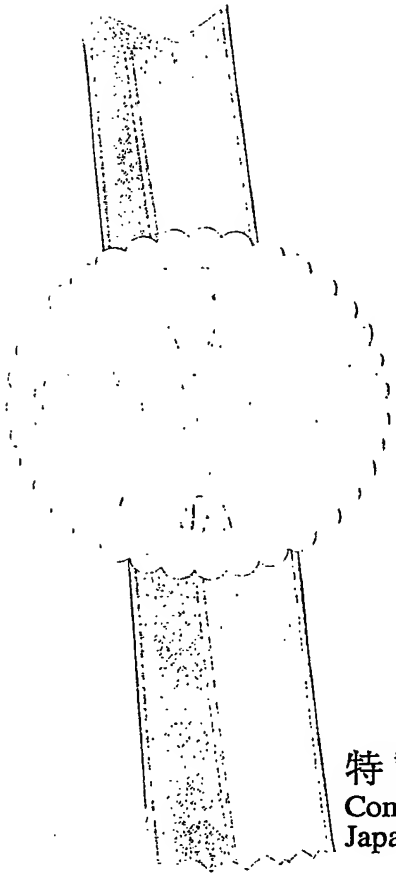
This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日
Date of Application: ２００３年１０月１０日

出願番号
Application Number: 特願２００３－３５２９１３

[ST. 10/C]: [ＪＰ２００３－３５２９１３]

出願人
Applicant(s): 松下電器産業株式会社



PRIORITY DOCUMENT
SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH
RULE 17.1(a) OR (b)

２００４年１１月１２日

特許庁長官
Commissioner,
Japan Patent Office

小川洋

出証番号　出証特２００４－３１０２３３３

【書類名】　特許願
【整理番号】　2131150541
【提出日】　平成15年10月10日
【あて先】　特許庁長官　殿
【国際特許分類】　G11B　7/00
【発明者】
　【住所又は居所】　大阪府門真市大字門真１００６番地　松下電器産業株式会社内
　【氏名】　池田　航
【発明者】
　【住所又は居所】　大阪府門真市大字門真１００６番地　松下電器産業株式会社内
　【氏名】　岩本　啓明
【発明者】
　【住所又は居所】　大阪府門真市大字門真１００６番地　松下電器産業株式会社内
　【氏名】　岡田　智之
【特許出願人】
　【識別番号】　000005821
　【氏名又は名称】　松下電器産業株式会社
【代理人】
　【識別番号】　100090446
　【弁理士】
　【氏名又は名称】　中島　司朗
【手数料の表示】
　【予納台帳番号】　014823
　【納付金額】　21,000円
【提出物件の目録】
　【物件名】　特許請求の範囲　1
　【物件名】　明細書　1
　【物件名】　図面　1
　【物件名】　要約書　1
　【包括委任状番号】　9003742

【請求項１】
映像データとプログラムが記録されており、
映像データの管理区間毎に、前記プログラムの状態を制御する管理情報が記録されている
ことを特徴とする情報記録媒体。
【請求項２】
前記管理区間とは、前記映像データをシナリオごとに分割したタイトル、情報記録媒体全体、複数の情報記録媒体の何れかである
ことを特徴とする請求項１記載の情報記録媒体。
【請求項３】
前記管理区間とともに、前記管理区間の一部を指定できる前記管理情報が記録されている請求項２記載の情報記録媒体。
【請求項４】
前記管理区間ごとの制御とは、前記管理区間終了時に前記プログラムを終了させることである、請求項１記載の情報記録媒体。
【請求項５】
前記管理区間ごとの制御とは、前記管理区間開始時に前記プログラムを起動させることである、請求項１記載の情報記録媒体。
【請求項６】
前記管理情報は、前記管理区間開始時に前記プログラムを起動するかしないか識別するための情報を有する
ことを特徴とする請求項１記載の情報記録媒体。
【請求項７】
前記管理情報が、前記管理区間開始時に前記プログラムを一時停止するか再開するか識別するための情報を有する
ことを特徴とする請求項１記載の情報記録媒体。
【請求項８】
２つ以上の前記管理情報には、同じプログラムに関する管理情報が含まれており、
前記管理情報は、一の管理区間から他の管理区間への遷移を制御する情報であり、
当該管理情報により制御される遷移とは、
プログラムの終了及び再起動を介さず、遷移前の管理区間でのプログラムの状態を遷移後の管理区間に継続させることである、請求項１記載の情報記録媒体。
【請求項９】
前記プログラムとはイベントを受けて駆動するイベントハンドラを備えていることを特徴とする請求項１記載の情報記録媒体。
【請求項１０】
前記イベントとはユーザー操作がなされたことを示すイベント、
前記映像データの所定の時点に達したことを示すイベント、
機器の状態が変化したことを示すイベント
の何れかであることを特徴とする請求項９記載の情報記録媒体。
【請求項１１】
前記イベントとは管理単位の終了を示すイベントであり、イベントに対応したプロセスがプログラムを終了させる
ことを特徴とした請求項１０記載の情報記録媒体。
【請求項１２】
前記イベントとは管理単位の開始を示すイベントと終了を示すイベントであり、
終了を示すイベントに対応したプロセスがプログラムを一時停止させ、開始を示すイベントに対応したプロセスがプログラムを再開させる
ことを特徴とした請求項１０記載の情報記録媒体。

【請求項１３】
映像データとプログラム、及び、前記プログラムが参照するデータが記録されており、
管理区間開始時に前記プログラムをメモリ上に読み込み、前記管理区間終了時にメモリ上に読み込まれた前記プログラムを解放するための管理情報が記録されている
ことを特徴とする情報記録媒体。
【請求項１４】
前記管理区間とは、前記映像データをシナリオごとに分割したタイトル、情報記録媒体全体、複数の情報記録媒体の何れかである
ことを特徴とする請求項１３記載の情報記録媒体。
【請求項１５】
前記管理区間とともに、前記管理区間の一部を指定できる前記管理情報が記録されている、請求項１４記載の情報記録媒体。
【請求項１６】
２つ以上の前記管理情報は、同じプログラム及び前記プログラムが参照するデータに関する管理情報であり、
前記管理情報は、一の管理区間から他の管理区間への遷移を制御する情報であり、
当該管理情報により制御される遷移とは、
プログラムの終了及び再起動、並びに、データの解放及び読み込みを介さず、遷移前の管理区間でのプログラム及びデータを保持したまま遷移後の管理区間に継続させることである、請求項１記載の情報記録媒体。
【請求項１７】
前記管理情報はメモリ上にデータ読み込むための優先度情報を備える、請求項１３記載の情報記録媒体。
【請求項１８】
前記優先度情報とは、メモリに必ず読み込まなければならないデータと必ずしも読み込まなくてもよいデータとを区別する
ことを特徴とする請求項１８記載の情報記録媒体。
【請求項１９】
前記管理情報とは情報記録媒体上のファイルとメモリ上のファイル情報を備えていることを特徴とする請求項１３記載の情報記録媒体。
【請求項２０】
前記管理情報は、メモリ上の複数のファイルを情報記録媒体上の１つのファイルとして対応づけて管理できることを特徴とする請求項１９記載の情報記録媒体。
【請求項２１】
請求項１記載の管理情報と請求項１３記載の管理情報において、前記管理区間が同じ区間である管理データを結合することにより、プログラムの状態とデータの読み込みを制御することが可能な管理情報を備えたことを特徴とする情報記録媒体。
【請求項２２】
請求項２１記載の管理情報が請求項６又は７記載の識別情報を備えており、データ読み込みの優先度は前記識別情報に対応して一意に決めることができることを特徴とする情報記録媒体。
【請求項２３】
前記プログラムが前記映像データとは別ファイルとして、別の場所に記録されていることを特徴とした請求項１記載の情報記録媒体。
【請求項２４】
前記プログラムが前記映像データとともに多重化されていることを特徴とした請求項１記載の情報記録媒体。
【請求項２５】
前記プログラムが前記映像データの途中にインターリーブ配置されて記録されていることを特徴とした請求項１記載の情報記録媒体。

【請求項２６】
前記管理情報による管理及び制御は、
請求項２３～請求項２５の何れかに記載のプログラムを対象としている
ことを特徴とする、請求項２１記載の情報記録媒体。
【請求項２７】
請求項２１記載の管理情報において、同じ前記管理単位内で同時に起動されないプログラム同士をグループ化して管理する
ことを特徴とする情報記録媒体。
【請求項２８】
前記管理情報が映像データとは別に管理されている
ことを特徴とする請求項１，１３，２１の何れかに記載の情報記録媒体。
【請求項２９】
前記管理情報が映像データと同じファイルとして管理されている
ことを特徴とする請求項１，１３，２１の何れかに記載の情報記録媒体。
【請求項３０】
前記プログラムの管理情報とは別に、前記映像データの再生制御情報が記録されている
ことを特徴とした情報記録媒体。
【請求項３１】
映像データとプログラムが記録された情報記録媒体を再生する再生装置であって、
前記映像データを再生する再生部と、
前記プログラムを処理するプログラム処理部と、
管理情報にしたがって管理区間ごとに前記プログラムの状態を制御する手段と　を備えることを特徴とする再生装置。
【請求項３２】
前記プログラムの状態を制御する手段とは、プログラムの起動あるいはプログラムの終了、あるいはプログラムの一時停止と再開の何れかである
ことを特徴とする再生装置。
【請求項３３】
前記管理情報とは、請求項１記載の管理情報である
ことを特徴とする請求項３１記載の再生装置。
【請求項３４】
映像データとプログラムが記録された情報記録媒体を再生する再生装置であって、
前記プログラムをメモリに読み込む手段とメモリから解放する手段を備え、管理情報にしたがって管理区間ごとにデータを制御する
ことを特徴とする再生装置。
【請求項３５】
前記管理情報とは、請求項１３記載の管理情報であることを特徴とする請求項３４記載の再生装置。
【請求項３６】
請求項３１記載のプログラムの状態を制御する手段とは、
管理区間開始時にプログラムを起動する手段、
管理区間終了時にプログラムを終了する手段、
イベントによりプログラムを起動する手段、
イベントによりプログラムを終了する手段、
プログラムからの通知により別のプログラムを起動する手段、
プログラムからの通知により別のプログラムを終了する手段
の何れか、あるいは、いくつかの手段であることを特徴とする再生装置。
【請求項３７】
前記プログラムに終了通知を行っても前記プログラムが自動的に終了しない場合、前記プログラムの状態を制御する手段がプログラムを強制的に終了させる手段

を備えることを特徴とする請求項３１記載の再生装置。
【請求項３８】
前記メモリ上のプログラムをプログラム処理部に読み込む手段とプログラム処理部からプログラムを解放する手段
を備えたことを特徴とする請求項３４記載の再生装置。
【請求項３９】
請求項３４記載の再生装置であって、前記メモリにプログラムを読み込めるか否かを、プログラムのサイズとメモリの空き容量を比較して判定する手段と、読み込めると判定した場合に一つ又は複数のプログラムを選択してメモリに読み込む手段を備える
ことを特徴とする再生装置。
【請求項４０】
前記判定手段として、請求項１７記載の優先度情報の順で前記プログラムのサイズとメモリの空き容量を比較を行う
ことを特徴とする請求項３９記載の再生装置。
【請求項４１】
請求項３８記載の再生装置において、前記メモリ内にデータが読み込まれていないため、前記プログラム処理部にプログラムを読み込めない場合、データの読み込みに失敗する手段、
タイマーを設定して指定時間まで読み込み待ちを行い、指定時間が過ぎてもデータが読み込めないときに読み込みに失敗する手段、
データが読み込めるまで待ち続ける手段
の何れかを備えたことを特徴とする再生装置。
【請求項４２】
請求項３４記載の再生装置であり、前記映像データを読み込み中であっても、前記プログラムを前記メモリに読み込む手段を
備えたことを特徴とする再生装置。
【請求項４３】
前記映像データは再生制御部で制御され、
再生制御部は、
前記プログラム処理部による処理にて前記プログラムが異常状態になった場合でも、前記映像データの再生を保証する
ことを特徴とする請求項３１記載の再生装置。
【請求項４４】
請求項１７記載の優先度情報により前記プログラムの読み込みを判定する方法であって、
優先度を判定し優先度順に処理を行うステップと、
前記プログラムのサイズとメモリの空き容量を比較するステップと、
前記メモリの空き容量が前記プログラムサイズより大きい場合は前記プログラムを読み込むステップと
を備えることを特徴とする再生方法。

【書類名】明細書
【発明の名称】情報記録媒体、その再生装置及び再生方法
【技術分野】
【０００１】
本発明は、ＢＤ－ＲＯＭ等、映像データを記録した情報記録媒体、その再生装置及び再生方法に関し、特に、映像データとプログラムとを含むコンテンツの開発にプログラミング環境を導入する場合の改良に関する。
【背景技術】
【０００２】
映像データを記録した情報記録媒体の代表格は、ＤＶＤ（以下、ＳＤ－ＤＶＤまたは単にＤＶＤと称する）である。以降、従来のＤＶＤについて説明する。
図１は、ＳＤ－ＤＶＤの構造を示した図である。図１の下段に示すように、ＤＶＤディスク上にはリードインからリードアウトまでの間に論理アドレス空間が設けられ、論理アドレス空間の先頭からファイルシステムのボリューム情報が記録され、続いて映像音声などのアプリケーションデータが記録されている。
【０００３】
ファイルシステムとは、ＩＳＯ９６６０やＵＤＦ（Ｕｎｉｖｅｒｓａｌ　Ｄｉｓｃ　Ｆｏｒｍａｔ）のことであり、ディスク上のデータをディレクトリまたはファイルと呼ばれる単位で表現する仕組みである。日常使っているＰＣ（パーソナルコンピュータ）の場合でも、ＦＡＴまたはＮＴＦＳと呼ばれるファイルシステムを通すことにより、ディレクトリやファイルという構造でハードディスクに記録されたデータがコンピュータ上で表現され、ユーザビリティを高めている。　ＳＤ－ＤＶＤの場合、ＵＤＦ及びＩＳＯ９６６０両方を使用しており（両方を合わせて「ＵＤＦブリッジ」と呼ぶ事がある）、ＵＤＦまたはＩＳＯ９６６０どちらのファイルシステムドライバによってもデータの読み出し（ここで取り扱うＤＶＤはパッケージメディア用のＲＯＭディスクであり、物理的に書き込みが不可能である）ができるようになっている。
【０００４】
ＤＶＤ上に記録されたデータは、ＵＤＦブリッジを通して、図１左上に示すようなディレクトリまたはファイルとして見ることができる。ルートディレクトリ（図中「ＲＯＯＴ」）の直下に「ＶＩＤＥＯ＿ＴＳ」と呼ばれるディレクトリが置かれ、ここにＤＶＤのアプリケーションデータが記録されている。アプリケーションデータは、複数のファイルとして記録され、主なファイルとして以下のものがある。

ＶＩＤＥＯ＿ＴＳ．ＩＦＯ	ディスク再生制御情報ファイル
ＶＴＳ＿０１＿０．ＩＦＯ	ビデオタイトルセット＃１再生制御情報ファイル
ＶＴＳ＿０１＿０．ＶＯＢ	ビデオタイトルセット＃１ストリームファイル
・・・・・	

拡張子として２つの種類が存在する。「ＩＦＯ」は再生制御情報が記録されたファイルであって、「ＶＯＢ」はＡＶデータであるＭＰＥＧストリームが記録されたファイルである。再生制御情報とは、ＤＶＤで採用されたインタラクティビティ（ユーザの操作に応じて再生を動的に変化させる技術）を実現するための情報や、メタデータのようなタイトルやＡＶストリームに付属する情報などのことである。また、ＤＶＤでは一般的に再生制御情報のことをナビゲーション情報と呼ぶことがある。
【０００５】
再生制御情報ファイルは、ディスク全体を管理する「ＶＩＤＥＯ＿ＴＳ．ＩＦＯ」と、個々のビデオタイトルセット（ＤＶＤでは複数のタイトル、言い換えれば異なる映画や異なるバージョンの映画を１枚のディスクに記録することが可能である。）毎の再生制御情報である「ＶＴＳ＿０１＿０．ＩＦＯ」がある。ここで、ファイル名ボディにある「０１」はビデオタイトルセットの番号を示しており、例えば、ビデオタイトルセット＃２の場

合は、「ＶＴＳ＿０２＿０．ＩＦＯ」となる。

【０００６】

図１の右上部は、ＤＶＤのアプリケーション層でのＤＶＤナビゲーション空間であり、前述した再生制御情報が展開された論理構造空間である。「ＶＩＤＥＯ＿ＴＳ．ＩＦＯ」内の情報は、ＶＭＧＩ（ＶＩＤＥＯ　Ｍａｎａｇｅｒ　Ｉｎｆｏｒｍａｔｉｏｎ）として、「ＶＴＳ＿０１＿０．ＩＦＯ」または、他のビデオタイトルセット毎に存在する再生制御情報はＶＴＳＩ（Ｖｉｄｅｏ　Ｔｉｔｌｅ　Ｓｅｔ　Ｉｎｆｏｒｍａｔｉｏｎ）としてＤＶＤナビゲーション空間に展開される。

【０００７】

ＶＴＳＩの中にはＰＧＣ（Ｐｒｏｇｒａｍ　Ｃｈａｉｎ）と呼ばれる再生シーケンスの情報であるＰＧＣＩ（Ｐｒｏｇｒａｍ　Ｃｈａｉｎ　Ｉｎｆｏｒｍａｔｉｏｎ）が記述されている。ＰＧＣＩは、Ｃｅｌｌの集合とコマンドと呼ばれる一種のプログラミング情報によって構成されている。Ｃｅｌｌ自身はＶＯＢ（Ｖｉｄｅｏ　Ｏｂｊｅｃｔの略であり、ＭＰＥＧストリームを指す）の一部区間または全部区間の集合であり、Ｃｅｌｌの再生は、当該ＶＯＢのＣｅｌｌによって指定された区間を再生することを意味している。

【０００８】

コマンドは、ＤＶＤの仮想マシンによって処理されるものであり、ブラウザ上で実行されるＪａｖａ（登録商標）スクリプトなどに近いものである。しかしながらＪａｖａ（登録商標）スクリプトが論理演算の他にウィンドウやブラウザの制御（例えば、新しいブラウザのウィンドを開くなど）を行うのに対して、ＤＶＤのコマンドは、論理演算の他にＡＶタイトルの再生制御、例えば、再生するチャプタの指定などを実行するだけのものである点で異なっている。

【０００９】

Ｃｅｌｌはディスク上に記録されているＶＯＢの開始及び終了アドレス（論理アドレス）をその内部情報として有しており、プレーヤは、Ｃｅｌｌに記述されたＶＯＢの開始及び終了アドレス情報を使ってデータの読み出し、再生を実行する。

図１はＡＶストリーム中に埋め込まれているナビゲーション情報を説明する概略図である。ＳＤ－ＤＶＤの特長であるインタラクティビティは前述した「ＶＩＤＥＯ＿ＴＳ．ＩＦＯ」や「ＶＴＳ＿０１＿０．ＩＦＯ」などに記録されているナビゲーション情報だけによって実現されているのではなく、幾つかの重要な情報はナビゲーション・パック（ナビパックまたは、ＮＶ＿ＰＣＫと称する）と呼ばれる専用キャリアを使いＶＯＢ内に映像、音声データと一緒に多重化されている。

【００１０】

ここでは簡単なインタラクティビティの例としてメニューを説明する。メニュー画面上には、幾つかのボタンが現れ、夫々のボタンには当該ボタンが選択実行された時の処理が定義されている。また、メニュー上では一つのボタンが選択されており（ハイライトによって選択ボタン上に半透明色がオーバーレイされている）、ユーザは、リモコンの上下左右キーを使って、選択状態のボタンを上下左右の何れかのボタンに移動させることが出来る。リモコンの上下左右キーを使って、選択実行したいボタンまでハイライトを移動させ、決定する（決定キーを押す）ことによって対応するコマンドのプログラムが実行される。一般的には対応するタイトルやチャプタの再生がコマンドによって実行されている。

【００１１】

図２の左上部はＮＶ＿ＰＣＫ内の概要を示している。

ＮＶ＿ＰＣＫ内には、ハイライトカラー情報と個々のボタン情報などが含まれている。ハイライトカラー情報には、カラーパレット情報が記述され、オーバーレイ表示されるハイライトの半透明色が指定される。ボタン情報には、個々のボタンの位置情報である矩形領域情報と、当該ボタンから他のボタンへの移動情報（ユーザの上下左右キー操作夫々に対応する移動先ボタンの指定）と、ボタンコマンド情報（当該ボタンが決定された時に実行されるコマンド）が記述されている。

【００１２】

メニュー上のハイライトは、図２の中央右上部に示すように、オーバーレイ画像として作られる。オーバーレイ画像は、ボタン情報の矩形領域情報にカラーパレット情報の色をつけた物である。このオーバーレイ画像は図４６の右部に示す背景画像と合成されて画面上に表示される。

上述のようにして、ＤＶＤではメニューを実現している。また、何故、ナビゲーションデータの一部をＮＶ＿ＰＣＫを使ってストリーム中に埋め込んでいるのは、ストリームと同期して動的にメニュー情報を更新、例えば、映画再生中の途中５分～１０分の間にだけメニューが表示されるなど、同期タイミングが問題となりやすいアプリケーションの場合でも、問題なく実現できるようにしたためである。

【００１３】

図３は、ＤＶＤのＶＯＢのイメージである。図に示すように、映像、音声、字幕などのデータ（Ａ段）は、ＭＰＥＧシステム（ＩＳＯ／ＩＥＣ１３８１８－１）規格に基づいて、パケット及びパック化し（Ｂ段）、夫々を多重化して１本のＭＰＥＧプログラムストリームにしている（Ｃ段）。また、前述した通りインタラクティブを実現するためのボタンコマンドを含んだＮＶ＿ＰＣＫも一緒に多重化をされている。

【００１４】

ＭＰＥＧシステムの多重化の特徴は、多重化する個々のデータは、そのデコード順に基づくビット列になっているが、多重化されるデータ間、即ち、映像、音声、字幕の間は必ずしも再生順、言い換えればデコード順に基づいてビット列が形成されているわけではない。これはＭＰＥＧシステムストリームのデコーダモデル（一般にＳｙｓｔｅｍ　Ｔａｒｇｅｔ　Ｄｅｃｏｄｅｒ、またはＳＴＤと呼ばれる（図３のＤ段））が多重化を解いた後に個々のエレメンタリーストリームに対応するデコーダバッファを持ち、デコードタイミングまでに一時的にデータを蓄積している事に由来している。このデコーダバッファは、個々のエレメンタリーストリーム毎にサイズが異なり、映像に対しては、２３２ｋＢ、音声に対しては４ｋＢ、字幕に対しては５２ｋＢを夫々有している。このため、各デコーダバッファへのデータ入力タイミングは個々のエレメンタリストリームで異なるため、ＭＰＥＧシステムストリームとしてビット列を形成する順番と表示（デコード）されるタイミングにずれが生じている。

【００１５】

即ち、映像データと並んで多重化されている字幕データが必ずしも同一タイミングでデコードされているわけでは無い。

かかるＤＶＤの構成は、以下の特許文献１に記載されている。

【特許文献１】特許第2813245号

【発明の開示】

【発明が解決しようとする課題】

【００１６】

ところで映像データと、プログラムとを含むコンテンツを、情報記録媒体で配布しようとすると、プログラムの状態がおかしくなることによりプレーヤ自体の動作やＡＶ再生が破綻していまう恐れがある。次世代の情報記録媒体であるＢＤ－ＲＯＭは、上述したようなナビゲーションコマンドではなく、Ｊａｖａ（登録商標）プログラムを含むコンテンツの記録が予定されており、かかるコンテンツにあたっては、プレーヤの動作保障が深刻な問題になる。

【００１７】

メモリなどのリソースを厳密に管理するため、従来ＤＶＤプレーヤ規格では、リソースを管理する際にメモリリソースのサイズを固定化している。しかしメモリサイズの固定化は、プレーヤ設計の自由度を失わせるものであり、アプリケーションが発展してもプレーヤは、その発展から取り残されてしまうという問題点がある。

本発明の目的は、プレーヤの動作保障を実現しつつも、プレーヤ設計の自由度を高めることができる情報記録媒体を提供することである。

【課題を解決するための手段】

【００１８】
上記課題を解決するため本発明にかかる情報記録媒体は、映像データとプログラムが記録されており、管理区間ごとに前記プログラムの状態を制御する管理情報が記録されていることを特徴としている。
【発明の効果】
【００１９】
上述した構成では、タイトルなどを管理単位としてアプリケーションの起動及び終了やデータの読み込み及び解放のタイミングが管理可能になるので、管理単位内で動作するアプリケーションを容易に管理でき、管理単位外に影響を及ぼさない仕組みを提供することが可能となる。
ここで前記管理情報はメモリ上にデータ読み込むための優先度情報を備えていてもよい。管理情報としてデータ読み込み優先度を用いることにより、プレーヤに搭載されているメモリサイズに応じて読み込むデータサイズを調整することが可能となり、メモリサイズが小さい場合でもアプリケーションは動作保証され、メモリサイズが大きい場合はアプリケーションの動作や切替が早くなる利便性を提供することが可能となる。
【発明を実施するための最良の形態】
【００２０】
（実施例１）
まず最初に本発明の第１の実施の形態について説明する。
（ディスク上の論理データ構造）
図４は、ＢＤ－ＲＯＭ（以降、「ＢＤ」と称する場合もある）の構成、特にディスク媒体であるＢＤディスク（１０４）と、ディスクに記録されているデータ（１０１、１０２、１０３）の構成を示す図である。ＢＤディスク（１０４）に記録されるデータは、ＡＶデータ（１０３）と、ＡＶデータに関する管理情報及びＡＶ再生シーケンスなどのＢＤ管理情報（１０２）と、インタラクティブを実現するＢＤ再生プログラム（１０１）である。本実施の形態では、映画などのＡＶコンテンツを再生するためのＡＶアプリケーションを主眼においてのＢＤディスクの説明を行うが、ＢＤディスクをＣＤ－ＲＯＭやＤＶＤ－ＲＯＭの様にコンピュータ用途の記録媒体としてしようすることも当然のことながら可能である。　図５は、上述したＢＤディスクに記録されている論理データを示した図である。ＢＤディスクは、他の光ディスク、例えばＤＶＤやＣＤなどと同様にその内周から外周に向けてらせん状に記録領域を持ち、内周のリード・インと外周のリード・アウトの間に論理データを記録できる論理アドレス空間を有している。また、リード・インの内側にはＢＣＡ（Ｂｕｒｓｔ　Ｃｕｔｔｉｎｇ　Ａｒｅａ）と呼ばれるドライブでしか読み出せない特別な領域がある。この領域はアプリケーションから読み出せないため、例えば著作権保護技術などに利用されることがよくある。
【００２１】
論理アドレス空間には、ファイルシステム情報（ボリューム）を先頭に映像データなどのアプリケーションデータが記録されている。ファイルシステムとは従来技術で説明した通り、ＵＤＦやＩＳＯ９６６０などのことであり、通常のＰＣと同じように記録されている論理データをディレクトリ、ファイル構造を使って読み出しする事が可能になっている。
【００２２】
本実施例の場合、ＢＤディスク上のディレクトリ、ファイル構造は、ルートディレクトリ（ＲＯＯＴ）直下にＢＤＶＩＤＥＯディレクトリが置かれている。このディレクトリはＢＤ－ＲＯＭで扱うＡＶコンテンツや管理情報などのデータ（図４で説明した１０１、１０２、１０３）が記録されているディレクトリである。

ＢＤＶＩＤＥＯディレクトリの下には、次の７種類のファイルが記録されている。

ＢＤ．ＩＮＦＯ（ファイル名固定）

「ＢＤ管理情報」の一つであり、ＢＤディスク全体に関する情報を記録したファイルである。ＢＤプレーヤは最初にこのファイルを読み出す。

ＢＤ．ＰＲＯＧ（ファイル名固定）
「ＢＤ再生プログラム」の一つであり、ＢＤディスク全体に関わるプログラムを記録したファイルである。

ＸＸＸ．ＰＬ（「ＸＸＸ」は可変、拡張子「ＰＬ」は固定）
「ＢＤ管理情報」の一つであり、シナリオを記録するプレイリスト（Ｐｌａｙ　Ｌｉｓｔ）情報を記録したファイルである。プレイリスト毎に１つのファイルを持っている。

ＸＸＸ．ＰＲＯＧ（「ＸＸＸ」は可変、拡張子「ＰＬ」は固定）
「ＢＤ再生プログラム」の一つであり、前述したプレイリスト毎のプログラムを記録したファイルである。プレイリストとの対応はファイルボディ名（「ＸＸＸ」が一致する）によって識別される。

ＹＹＹ．ＶＯＢ（「ＹＹＹ」は可変、拡張子「ＶＯＢ」は固定）
「ＡＶデータ」の一つであり、ＶＯＢ（従来例で説明したＶＯＢと同じ）を記録したファイルである。ＶＯＢ毎に１つのファイルを持っている。

ＹＹＹ．ＶＯＢＩ（「ＹＹＹ」は可変、拡張子「ＶＯＢＩ」は固定）
「ＢＤ管理情報」の一つであり、ＡＶデータであるＶＯＢに関わる管理情報を記録したファイルである。ＶＯＢとの対応はファイルボディ名（「ＹＹＹ」が一致する）によって識別される。

ＺＺＺ．ＰＮＧ（「ＺＺＺ」は可変、拡張子「ＰＮＧ」は固定）
「ＡＶデータ」の一つであり、字幕及びメニューを構成するためのイメージデータＰＮＧ（Ｗ３Ｃによって標準化された画像フォーマットであり「ピング」と読む）を記録したファイルである。１つのＰＮＧイメージ毎に１つのファイルを持つ。

（プレーヤの構成）
次に、前述したＢＤディスクを再生するプレーヤの構成について図６及び図７を用いて説明する。

【００２３】
図６は、プレーヤの大まかな機能構成を示すブロック図である。
ＢＤディスク（２０１）上のデータは、光ピックアップ（２０２）を通して読み出される。読み出されたデータは夫々のデータの種類に応じて専用のメモリに記録される。ＢＤ再生プログラム（「ＢＤ．ＰＲＯＧ」または「ＸＸＸ．ＰＲＯＧ」ファイルの中身）はプログラム記録メモリ（２０３）に、ＢＤ管理情報（「ＢＤ．ＩＮＦＯ」、「ＸＸＸ．ＰＬ」または「ＹＹＹ．ＶＯＢＩ」）は管理情報記録メモリ（２０４）に、ＡＶデータ（「ＹＹＹ．ＶＯＢ」または「ＺＺＺ．ＰＮＧ」）はＡＶ記録メモリ（２０５）に夫々記録される。

【００２４】
プログラム記録メモリ（２０３）に記録されたＢＤ再生プログラムはプログラム処理部（２０６）によって、管理情報記録メモリ（２０４）に記録されたＢＤ管理情報は管理情報処理部（２０７）によって、また、ＡＶ記録メモリ（２０５）に記録されたＡＶデータはプレゼンテーション処理部（２０８）によって夫々処理される。
プログラム処理部（２０６）は、管理情報処理部（２０７）より再生するプレイリストの情報やプログラムの実行タイミングなどのイベント情報を受け取りプログラムの処理を行う。また、プログラムでは再生するプレイリストを動的に変える事が可能であり、この場合は管理情報処理部（２０７）に対してプレイリストの再生命令を送ることで実現する

。プログラム処理部（２０６）は、ユーザからのイベント、即ちリモコンキーからのリクエストを受け、ユーザイベントに対応するプログラムがある場合は、実行処理する。

【００２５】

管理情報処理部（２０７）は、プログラム処理部（２０６）の指示を受け、対応するプレイリスト及びプレイリストに対応したＶＯＢの管理情報を解析し、プレゼンテーション処理部（２０８）に対象となるＡＶデータの再生を指示する。また、管理情報処理部（２０７）は、プレゼンテーション処理部（２０８）より基準時刻情報を受け取り、時刻情報に基づいてプレゼンテーション処理部（２０８）にＡＶデータ再生の停止指示を行い、また、プログラム処理部（２０６）に対してプログラム実行タイミングを示すイベントを生成する。

【００２６】

プレゼンテーション処理部（２０８）は、映像、音声、字幕／イメージ夫々に対応するデコーダを持ち、管理情報処理部（２０７）からの指示に従い、ＡＶデータのデコード及び出力を行う。映像データ及び字幕／イメージの場合は、デコード後に夫々の専用プレーン、ビデオプレーン（２１０）及びイメージプレーン（２０９）に描画され、合成処理部（２１１）によって映像の合成処理が行われＴＶなどの表示デバイスへ出力される。

【００２７】

図６で示すように、ＢＤプレーヤは図４で示したＢＤディスクに記録されているデータ構成に基づいた構成をとっている。

図７は前述したプレーヤ構成を詳細化したブロック図である。図７では、ＡＶ記録メモリ（２０５）はイメージメモリ（３０８）とトラックバッファ（３０９）に、プログラム処理部（２０６）はプログラムプロセッサ（３０２）とＵＯＰマネージャ（３０３）に、管理情報処理部（２０７）はシナリオプロセッサ（３０５）とプレゼンテーションコントローラ（３０６）に、プレゼンテーション処理部（２０８）はクロック（３０７）、デマルチプレクサ（３１０）、イメージプロセッサ（３１１）、ビデオプロセッサ（３１２）とサウンドプロセッサ（３１３）に夫々対応／展開している。

【００２８】

ＢＤディスク（２０１）から読み出されたＶＯＢデータ（ＭＰＥＧストリーム）はトラックバッファ（３０９）に、イメージデータ（ＰＮＧ）はイメージメモリ（３０８）に夫々記録される。デマルチプレクサ（３１０）がクロック（３０７）の時刻に基づき、トラックバッファ（３０９）に記録されたＶＯＢデータを抜き出し、映像データをビデオプロセッサ（３１２）に音声データをサウンドプロセッサ（３１３）に夫々送り込む。ビデオプロセッサ（３１２）及びサウンドプロセッサ（３１３）は夫々ＭＰＥＧシステム規格で定める通りに、デコーダバッファとデコーダから夫々構成されている。即ち、デマルチプレクサ（３１０）から送りこまれる映像、音声夫々のデータは、夫々のデコーダバッファに一時的に記録され、クロック（３０７）に従い個々のデコーダでデコード処理される。

【００２９】

イメージメモリ（３０８）に記録されたＰＮＧは、次の２つの処理方法がある。イメージデータが字幕用の場合は、プレゼンテーションコントローラ（３０６）によってデコードタイミングが指示される。クロック（３０７）からの時刻情報をシナリオプロセッサ（３０５）が一旦受け、適切な字幕表示が行えるように、字幕表示時刻（開始及び終了）になればプレゼンテーションコントローラ（３０６）に対して字幕の表示、非表示の指示を出す。プレゼンテーションコントローラ（３０６）からデコード／表示の指示を受けたイメージプロセッサ（３１１）は対応するＰＮＧデータをイメージメモリ（３０８）から抜き出し、デコードし、イメージプレーン（３１４）に描画する。

【００３０】

次に、イメージデータがメニュー用の場合は、プログラムプロセッサ（３０２）によってデコードタイミングが指示される。プログラムプロセッサ（３０２）が何時イメージのデコードを指示するかは、プログラムプロセッサ（３０２）が処理しているＢＤプログラ

ムに因るものであって一概には決まらない。
イメージデータ及び映像データは、図６で説明したように夫々デコード後にイメージプレーン（３１４）、ビデオプレーン（３１５）に記録され、合成処理部（３１６）によって合成出力される。
【００３１】
ＢＤディスク（２０１）から読み出された管理情報（シナリオ、ＡＶ管理情報）は、管理情報記録メモリ（３０４）に記録されるが、シナリオ情報（「ＢＤ．ＩＮＦＯ」及び「ＸＸＸ．ＰＬ」）はシナリオプロセッサ（３０５）によって読み出され処理される。また、ＡＶ管理情報（「ＹＹＹ．ＶＯＢＩ」）はプレゼンテーションコントローラ（３０６）によって読み出され処理される。
【００３２】
シナリオプロセッサ（３０５）は、プレイリストの情報を解析し、プレイリストによって参照されているＶＯＢとその再生位置をプレゼンテーションコントローラ（３０６）に指示し、プレゼンテーションコントローラ（３０６）は対象となるＶＯＢの管理情報（「ＹＹＹ．ＶＯＢＩ」）を解析して、対象となるＶＯＢを読み出すようにドライブコントローラ（３１７）に指示を出す。
【００３３】
ドライブコントローラ（３１７）はプレゼンテーションコントローラ（３０６）の指示に従い、光ピックアップを移動させ、対象となるＡＶデータの読み出しを行う。読み出されたＡＶデータは、前述したようにイメージメモリ（３０８）またはトラックバッファ（３０９）に記録される。
また、シナリオプロセッサ（３０５）は、クロック（３０７）の時刻を監視し、管理情報で設定されているタイミングでイベントをプログラムプロセッサ（３０２）に投げる。
【００３４】
プログラム記録メモリ（３０１）に記録されたＢＤプログラム（「ＢＤ．ＰＲＯＧ」または「ＸＸＸ．ＰＲＯＧ」）は、プログラムプロセッサ３０２によって実行処理される。プログラムプロセッサ（３０２）がＢＤプログラムを処理するのは、シナリオプロセッサ（３０５）からイベントが送られてきた場合か、ＵＯＰマネージャ（３０３）からイベントが送られたきた場合である。ＵＯＰマネージャ（３０３）は、ユーザからリモコンキーによってリクエストが送られてきた場合に、プログラムプロセッサ（３０２）にイベントを生成する。

（アプリケーション空間）
図８は、ＢＤ－ＲＯＭのアプリケーション空間をを示す図である。
【００３５】
ＢＤ－ＲＯＭのアプリケーション空間では、プレイリスト（ＰｌａｙＬｉｓｔ）が一つの再生単位になっている。プレイリストはセル（Ｃｅｌｌ）の再生シーケンスから構成される静的なシナリオと、プログラムによって記述される動的なシナリオを有している。プログラムによる動的なシナリオが無い限り、プレイリストは個々のセルを順に再生するだけであり、また、全てのセルの再生を終了した時点でプレイリストの再生は終了する。一方で、プログラムは、プレイリストを超えての再生記述や、ユーザ選択またはプレーヤの状態によって再生する対象を動的に変えることが可能である。典型的な例としてはメニューがあげられる。ＢＤ－ＲＯＭの場合、メニューとはユーザの選択によって再生するシナリオ、即ちプレイリストを動的に選択することである。
【００３６】
ここで言うプログラムは、時間イベントまたはユーザイベントによって実行されるイベントハンドラの事である。
時間イベントは、プレイリスト中に埋め込まれた時刻情報に基づいて生成されるイベントである。図７で説明したシナリオプロセッサ（３０５）からプログラムプロセッサ（３０２）に送られるイベントがこれに相当する。時間イベントが発行されると、プログラム

プロセッサ（３０２）はＩＤによって対応付けられるイベントハンドラを実行処理する。前述した通り、実行されるプログラムが他のプレイリストの再生を指示することが可能であり、この場合には、現在再生されているプレイリストの再生は中止され、指定されたプレイリストの再生へと遷移する。

【００３７】

ユーザイベントは、ユーザのリモコンキー操作によって生成されるイベントである。ユーザイベントは大きく２つのタイプに分けられる。一つ目は、カーソルキー（「上」「下」「左」「右」キー）または「決定」キーの操作によって生成されるメニュー選択のイベントである。メニュー選択のイベントに対応するイベントハンドラはプレイリスト内の限られた期間でのみ有効であり（プレイリストの情報として、個々のイベントハンドラの有効期間が設定されている）、リモコンの「上」「下」「左」「右」キーまたは「決定」キーが押された時に有効なイベントハンドラを検索して、有効なイベントハンドラがある場合は当該イベントハンドラが実行処理される。他の場合は、メニュー選択のイベントは無視されることになる。

【００３８】

二つ目のユーザイベントは、「メニュー」キーの操作によって生成されるメニュー呼び出しのイベントである。メニュー呼び出しのイベントが生成されると、グローバルイベントハンドラが呼ばれる。グローバルイベントハンドラはプレイリストに依存せず、常に有効なイベントハンドラである。この機能を使うことにより、ＤＶＤのメニューコール（タイトル再生中に音声、字幕メニューなどを呼び出し、音声または字幕を変更後に中断した地点からのタイトル再生を実行する）を実装することができる。

【００３９】

プレイリストで静的シナリオを構成する単位であるセル（Ｃｅｌｌ）はＶＯＢ（ＭＰＥＧストリーム）の全部または一部の再生区間を参照したものである。セルはＶＯＢ内の再生区間を開始、終了時刻の情報として持っている。個々のＶＯＢと一対になっているＶＯＢ管理情報（ＶＯＢＩ）は、その内部にタイムマップ（Ｔｉｍｅ　Ｍａｐ またはＴＭ）を有しており、このタイムマップによって前述したＶＯＢの再生、終了時刻をＶＯＢ内（即ち対象となるファイル「ＹＹＹ．ＶＯＢ」内）での読み出し開始アドレス及び終了アドレスを導き出すことが可能である。なおタイムマップの詳細は後述する。

（ＶＯＢの詳細）

図９は、本実施例で使用するＭＰＥＧストリーム（ＶＯＢ）の構成図である。図９に示すように、ＶＯＢは複数のＶＯＢＵ（Ｖｉｄｅｏ　Ｏｂｊｅｃｔ　Ｕｎｉｔ）によって構成されている。ＶＯＢＵは、ＭＰＥＧビデオストリームで言うＧＯＰ（Ｇｒｏｕｐ　Ｏｆ　Ｐｉｃｔｕｒｅｓ）を基準として、音声データも含んだ多重化ストリームとしての一再生単位である。ＶＯＢＵは０．４秒から１．０秒の時間を持ち、通常は０．５秒の再生時間を持っている。これはＭＰＥＧのＧＯＰの構造が通常は１５フレーム／秒（ＮＴＳＣの場合）によって導かれるものである。

【００４０】

ＶＯＢＵは、その内部にビデオパック（Ｖ＿ＰＣＫ）とオーディオパック（Ａ＿ＰＣＫ）を有している。各パックは１セクタ、本実施例の場合は２ｋＢ単位で構成されている。

図１０は、パックの構成を示した図である。

図１０に示すように、ビデオデータ及びオーディオデータといったエレメンタリデータは、ペイロードと呼ばれるパケットのデータ格納領域に先頭から順次入れられていく。ペイロードにはパケットヘッダが付けられ１つのパケットを構成する。パケットヘッダには、ペイロードに格納してあるデータがどのストリームなのか、ビデオなのかオーディオなのか、また、ビデオまたはオーディオが夫々複数ストリームある場合は、どのストリームのデータなのかを識別するためのＩＤ（ｓｔｒｅａｍ＿ｉｄ）と、当該ペイロードのデコード及び表示時刻情報であるタイムスタンプＤＴＳ及びＰＴＳが夫々記録されている。ＰＴＳ／ＤＴＳは必ずしも全てのパケットヘッダに記録されている訳ではなく、ＭＰＥＧに

よって記録するルールが規定されている。ルールの詳細についてはＭＰＥＧシステム（ＩＳＯ／ＩＥＣ１３８１８－１）規格書に記述されているので省略する。

【００４１】

パケットには更にヘッダ（パックヘッダ）が付けられ、パックを構成する。パックヘッダには、当該パックがいつデマルチプレクサを通過し、個々のエレメンタリストリームのデコーダバッファに入力されるかを示すタイムスタンプＳＣＲ（Ｓｙｓｔｅｍ　Ｃｌｏｃｋ　Ｒｅｆｅｒｅｎｃｅ）が記録されている。

（ＶＯＢのインターリーブ記録）

次に図１１及び図１２を用いてＶＯＢファイルのインターリーブ記録について説明する。

【００４２】

図１１上段は、前述したプレーヤ構成図の一部である。図の通り、ＢＤディスク上のデータは、光ピックアップを通してＶＯＢ即ちＭＰＥＧストリームであればトラックバッファへ入力され、ＰＮＧ即ちイメージデータであればイメージメモリへと入力される。

トラックバッファはＦＩＦＯであり、入力されたＶＯＢのデータは入力された順にデマルチプレクサへと送られる。この時、前述したＳＣＲに従って個々のパックはトラックバッファから引き抜かれデマルチプレクサを介してビデオプロセッサまたはサウンドプロセッサへとデータが送り届けられる。一方で、イメージデータの場合は、どのイメージを描画するかはプレゼンテーションコントローラによって指示される。また、描画に使ったイメージデータは、字幕用イメージデータの場合は同時にイメージメモリから削除されるが、メニュー用のイメージデータの場合は、イメージメモリ内にそのまま残される。これはメニューの描画はユーザ操作に依存するところがあるため、同一イメージを複数回描画する可能性があるためである。

【００４３】

図１１下段は、ＢＤディスク上でのＶＯＢファイル及びＰＮＧファイルのインターリーブ記録を示す図である。一般的にＲＯＭ、例えばＣＤ－ＲＯＭやＤＶＤ－ＲＯＭの場合、一連の連続再生単位となるＡＶデータは連続記録されている。これは、連続記録されている限り、ドライブは順次データを読み出しプレーヤ側に送り届けるだけで良いが、連続データが分断されてディスク上に離散配置されている場合は、個々の連続区間の間でシーク操作が入ることになり、この間データの読み出しが止まることになり、データの供給が止まる可能性があるからである。ＢＤ－ＲＯＭの場合も同様に、ＶＯＢファイルは連続領域に記録することができる方が望ましいが、例えば字幕データのようにＶＯＢに記録されている映像データと同期して再生されるデータがあり、ＶＯＢファイルと同様に字幕データも何らかの方法によってＢＤディスクから読み出す事が必要になる。

【００４４】

字幕データの読み出し方法の一手段として、ＶＯＢの再生開始前に一まとめで字幕用のイメージデータ（ＰＮＧファイル）を読み出してしまう方法がある。しかしながら、この場合には一時記録に使用する大量のメモリが必要となり、非現実的である。

そこで、本実施の形態では、ＶＯＢファイルを幾つかのブロックに分けて、イメージデータとインターリーブ記録する方式を使用している。図１１下段はそのインターリーブ記録を説明した図である。ＶＯＢファイルとイメージデータを適切にインターリーブ配置することで、前述したような大量の一時記録メモリ無しに、必要なタイミングでイメージデータをイメージメモリに格納することが可能になる。しかしながらイメージデータを読み出している際には、ＶＯＢデータの読み込みは当然のことながら停止することになる。

図１２は、この問題を解決するトラックバッファを使ったＶＯＢデータ連続供給モデルを説明する図である。

【００４５】

既に説明したように、ＶＯＢのデータは、一旦トラックバッファに蓄積される。トラックバッファへのデータ入力レートとトラックバッファからのデータ出力レートの間に差を設けると、ＢＤディスクからデータを読み出し続けている限り、トラックバッファのデータ蓄積量は増加をしていくことになる。ここでトラックバッファへの入力レートをＶａ、トラックバッファからの出力レートをＶｂとする。図１２の上段に記すようにＶＯＢの一連続記録領域が論理アドレスの”ａ１”から”ａ２”まで続くとする。”ａ２”から”ａ３”の間は、イメージデータが記録されていて、ＶＯＢデータの読み出しが行えない区間であるとする。

図１２の下段は、トラックバッファの内部を示す図である。横軸が時間、縦軸がトラックバッファ内部に蓄積されているデータ量を示している。時刻”ｔ１”がＶＯＢの一連続記録領域の開始点である”ａ１”の読み出しを開始した時刻を示している。この時刻以降、トラックバッファにはレートＶａ－Ｖｂでデータが蓄積されていくことになる。このレートは言うまでもなくトラックバッファの入出力レートの差である。時刻”ｔ２”は一連続記録領域の終了点である”ａ２”のデータを読み込む時刻である。即ち時刻”ｔ１”から”ｔ２”の間レートＶａ－Ｖｂでトラックバッファ内はデータ量が増加していき、時刻”ｔ２”でのデータ蓄積量はＢ（ｔ２）は下式によって求めることができる。

$$B(t2) = (Va - Vb) \times (t2 - t1) \qquad (式1)$$

この後、ＢＤディスク上のアドレス”ａ３”まではイメージデータが続くため、トラックバッファへの入力は０となり、出力レートである”－Ｖｂ”でトラックバッファ内のデータ量は減少していくことになる。これは読み出し位置”ａ３”まで、時刻でいう”ｔ３”までになる。

ここで大事なことは、時刻”ｔ３”より前にトラックバッファに蓄積されているデータ量が０になると、デコーダへ供給するＶＯＢのデータが無くなってしまい、ＶＯＢの再生がストップしてしまう可能性がある。しかしながら、時刻”ｔ３”でトラックバッファにデータが残っている場合には、ＶＯＢの再生がストップすることなく連続できることを意味している。

この条件は下式によって示すことができる。

$$B(t2) \geqq -Vb \times (t3 - t2) \qquad (式2)$$

即ち、式２を満たすようにイメージデータの配置を決めればよい事になる。

（ナビゲーションデータ構造）

図１３から図１９を用いて、ＢＤ－ＲＯＭのナビゲーションデータ（ＢＤ管理情報）構造について説明をする。図１３は、ＶＯＢ管理情報情報ファイル（”ＹＹＹ．ＶＯＢＩ”）の内部構造を示した図である。

【００４６】

ＶＯＢ管理情報は、当該ＶＯＢのストリーム属性情報（Ａｔｔｒｉｂｕｔｅ）とタイムマップ（ＴＭＡＰ）を有している。ストリーム属性は、ビデオ属性（Ｖｉｄｅｏ）、オーディオ属性（Ａｕｄｉｏ＃０～Ａｕｄｉｏ＃ｍ）個々に持つ構成となっている。特にオーディオストリームの場合は、ＶＯＢが複数本のオーディオストリームを同時に持つことができることから、オーディオストリーム数（Ｎｕｍｂｅｒ）によって、データフィールドの有無を示している。

下記はビデオ属性（Ｖｉｄｅｏ）の持つフィールドと夫々が持ち得る値である。

圧縮方式（Ｃｏｄｉｎｇ）：
　ＭＰＥＧ１
　ＭＰＥＧ２
　ＭＰＥＧ４

解像度（Ｒｅｓｏｌｕｔｉｏｎ）：
　１９２０ｘ１０８０
　１２８０ｘ７２０
　７２０ｘ４８０
　７２０ｘ５６５

アスペクト比（Ａｓｐｅｃｔ）
　４：３
　１６：９

フレームレート（Ｆｒａｍｅｒａｔｅ）
　６０
　５９．９４
　５０
　３０
　２９．９７
　２５
　２４

下記はオーディオ属性（Ａｕｄｉｏ）の持つフィールドと夫々が持ち得る値である。

圧縮方式（Ｃｏｄｉｎｇ）：
　ＡＣ３
　ＭＰＥＧ１
　ＭＰＥＧ２
　ＬＰＣＭ

チャンネル数（Ｃｈ）：
　１～８

言語属性（Ｌａｎｇｕａｇｅ）：

タイムマップ（ＴＭＡＰ）はＶＯＢＵ毎の情報を持つテーブルであって、当該ＶＯＢが有するＶＯＢＵ数（Ｎｕｍｂｅｒ）と各ＶＯＢＵ情報（ＶＯＢＵ＃１～ＶＯＢＵ＃ｎ）を持つ。個々のＶＯＢＵ情報は、ＶＯＢＵの再生時間長（Ｄｕｒａｔｉｏｎ）とＶＯＢＵのデータサイズ（Ｓｉｚｅ）を夫々有している。

図１４はＶＯＢＵ情報の詳細を説明する図である。

【００４７】

広く知られているように、ＭＰＥＧストリームは時間的側面とデータサイズとしての側面との２つを有している。例えば、音声の圧縮規格であるＡＣ３は固定ビットレートでの圧縮を行っているため、時間とアドレスとの関係は１次式によって求めることができる。しかしながらＭＰＥＧビデオデータの場合は、個々のフレームは固定の表示時間、例えばＮＴＳＣの場合は１フレームは１／２９．９７秒の表示時間を持つが、個々のフレームの

圧縮後のデータサイズは絵の特性や圧縮に使ったピクチャタイプ、いわゆるＩ／Ｐ／Ｂピクチャによってデータサイズは大きく変わってくる。従って、ＭＰＥＧビデオの場合は、時間とアドレスの関係は一般式の形で表現することは不可能である。

【００４８】

当然の事として、ＭＰＥＧビデオデータを多重化しているＭＰＥＧシステムストリーム、即ちＶＯＢも時間とデータとを一般式の形で表現することは不可能である。これに代わって、ＶＯＢ内での時間とアドレスとの関係を結びつけるのがタイムマップ（ＴＭＡＰ）である。図１４に示すように、各ＶＯＢＵ毎にＶＯＢＵ内のフレーム数と、ＶＯＢＵ内のパック数を夫々エントリーとして持つテーブルがタイムマップ（ＴＭＡＰ）である。

【００４９】

図１５を使って、タイムマップ（ＴＭＡＰ）の使い方を説明する。

図１５に示すように時刻情報が与えられた場合、先ずは当該時刻がどのＶＯＢＵに属するのかを検索する。これは、タイム亜マップのＶＯＢＵ毎のフレーム数を加算して行き、フレーム数の和が当該時刻を（フレーム数に換算して）超えるまたは一致するＶＯＢＵが当該ＶＯＢＵになる。次にタイムマップのＶＯＢＵ毎のサイズを当該ＶＯＢＵの直前のＶＯＢＵまで加算して行き、その値が与えられた時刻を含むフレームを再生するために読み出すべきパックの先頭アドレスになっている。

【００５０】

次に図１６を使って、プレイリスト情報（”ＸＸＸ．ＰＬ”）の内部構造を説明する。

プレイリスト情報は、セルリスト（ＣｅｌｌＬｉｓｔ）とイベントリスト（ＥｖｅｎｔＬｉｓｔ）から構成されている。

セルリスト（ＣｅｌｌＬｉｓｔ）は、プレイリスト内の再生セルシーケンスであり、本リストの記述順でセルが再生される事になる。セルリスト（ＣｅｌｌＬｉｓｔ）の中身は、セルの数（Ｎｕｍｂｅｒ）と各セル情報（Ｃｅｌｌ＃１～Ｃｅｌｌ＃ｎ）である。

【００５１】

セル情報（Ｃｅｌｌ＃）は、ＶＯＢファイル名（ＶＯＢＮａｍｅ）、当該ＶＯＢ内での有効区間開始時刻（Ｉｎ）及び有効区間終了時刻（Ｏｕｔ）と、字幕テーブル（ＳｕｂｔｉｔｌｅＴａｂｌｅ）を持っている。有効区間開始時刻（Ｉｎ）及び有効区間終了時刻（Ｏｕｔ）は、夫々当該ＶＯＢ内でのフレーム番号で表現され、前述したタイムマップ（ＴＭＡＰ）を使うことによって再生に必要なＶＯＢデータのアドレスを得る事ができる。

【００５２】

字幕テーブル（ＳｕｂｔｉｔｌｅＴａｂｌｅ）は、当該ＶＯＢと同期再生される字幕情報を持つテーブルである。字幕は音声同様に複数の言語を持つことができ、字幕テーブル（ＳｕｂｔｉｔｌｅＴａｂｌｅ）最初の情報も言語数（Ｎｕｍｂｅｒ）とそれに続く個々の言語ごとのテーブル（Ｌａｎｇｕａｇｅ＃１～Ｌａｎｇｕａｇｅ＃ｋ）から構成されている。

【００５３】

各言語のテーブル（Ｌａｎｇｕａｇｅ＃）は、言語情報（Ｌａｎｇｕａｇｅ）と、個々に表示される字幕の字幕情報数（Ｎｕｍｂｅｒ）と、個々に表示される字幕の字幕情報（Ｓｐｅｅｃｈ＃１～Ｓｐｅｅｃｈ＃ｊ）から構成され、字幕情報（Ｓｐｅｅｃｈ＃）は対応するイメージデータファイル名（Ｎａｍｅ）、字幕表示開始時刻（Ｉｎ）及び字幕表示終了時刻（Ｏｕｔ）と、字幕の表示位置（Ｐｏｓｉｔｉｏｎ）から構成されている。

【００５４】

イベントリスト（ＥｖｅｎｔＬｉｓｔ）は、当該プレイリスト内であげられるイベントを定義したテーブルである。イベントリストは、イベント数（Ｎｕｍｂｅｒ）に続いて個々のイベント（Ｅｖｅｎｔ＃１～Ｅｖｅｎｔ＃ｍ）から構成され、個々のイベント（Ｅｖｅｎｔ＃）は、イベントの種類（Ｔｙｐｅ）、イベントのＩＤ（ＩＤ）、イベント生成時刻（Ｔｉｍｅ）と有効期間（Ｄｕｒａｔｉｏｎ）から構成されている。

【００５５】

図１７は、個々のプレイリスト毎のイベントハンドラ（時間イベントと、メニュー選択

用のユーザイベント）を持つイベントハンドラテーブル（”ＸＸＸ．ＰＲＯＧ”）である。

イベントハンドラテーブルは、定義されているイベントハンドラ／プログラム数（Ｎｕｍｂｅｒ）と個々のイベントハンドラ／プログラム（Ｐｒｏｇｒａｍ＃１～Ｐｒｏｇｒａｍ＃ｎ）を有している。各イベントハンドラ／プログラム（Ｐｒｏｇｒａｍ＃）内の記述は、イベントハンドラ開始の定義（＜ｅｖｅｎｔ＿ｈａｎｄｌｅｒ＞タグ）と前述したイベントのＩＤと対になるイベントハンドラのＩＤ（ＩＤ）を持ち、その後に当該プログラムもＦｕｎｃｔｉｏｎに続く括弧”｛”と”｝”の間に記述する。

【００５６】

次に図１８を用いてＢＤディスク全体に関する情報（”ＢＤ．ＩＮＦＯ”）の内部構造について説明をする。

ＢＤディスク全体情報は、タイトルリスト（ＴｉｔｌｅＬｉｓｔ）とグローバルイベント用のイベントテーブル（ＥｖｅｎｔＴａｂｌｅ）から構成されている。

タイトルリスト（ＴｉｔｌｅＬｉｓｔ）は、ディスク内のタイトル数（Ｎｕｍｂｅｒ）と、これに続く各タイトル情報（Ｔｉｔｌｅ＃１～Ｔｉｔｌｅ＃ｎ）から構成されている。個々のタイトル情報（Ｔｉｔｌｅ）は、タイトルに含まれるプレイリストのテーブル（ＰＬＴａｌｂｌｅ）とタイトル内のチャプタリスト（ＣｈａｐｔｅｒＬｉｓｔ）を含んでいる。プレイリストのテーブル（ＰＬＴａｂｌｅ）はタイトル内のプレイリストの数（Ｎｕｍｂｅｒ）と、プレイリスト名（Ｎａｍｅ）即ちプレイリストのファイル名を有している。

【００５７】

チャプタリスト（ＣｈａｐｔｅｒＬｉｓｔ）は、当該タイトルに含まれるチャプタ数（Ｎｕｍｂｅｒ）と個々のチャプタ情報（Ｃｈａｐｔｅｒ＃１～Ｃｈａｐｔｅｒ＃ｎ）から構成され、チャプタ情報（Ｃｈａｐｔｅｒ＃）は当該チャプタが含むセルのテーブル（ＣｅｌｌＴａｂｌｅ）を持ち、セルのテーブル（ＣｅｌｌＴａｂｌｅ）はセル数（Ｎｕｍｂｅｒ）と個々のセルのエントリ情報（ＣｅｌｌＥｎｔｒｙ＃１～ＣｅｌｌＥｎｔｒｙ＃ｋ）から構成されている。セルのエントリ情報（ＣｅｌｌＥｎｔｒｙ＃）は当該セルを含むプレイリスト名と、プレイリスト内でのセル番号によって記述されている。

【００５８】

イベントリスト（ＥｖｅｎｔＬｉｓｔ）は、グローバルイベントの数（Ｎｕｍｂｅｒ）と個々のグローバルイベントの情報を持っている。ここで注意すべきは、最初に定義されるグローバルイベントは、ファーストイベント（ＦｉｒｓｔＥｖｅｎｔ）と呼ばれ、ＢＤディスクがプレーヤに挿入された時、最初に呼ばれるイベントである。グローバルイベント用イベント情報はイベントタイプ（Ｔｙｐｅ）とイベントのＩＤ（ＩＤ）だけを持っている。

【００５９】

図１９は、グローバルイベントハンドラのプログラムのテーブル（”ＢＤ．ＰＲＯＧ”）である。本テーブルは、図１７で説明したイベントハンドラテーブルと同一内容である。

（イベント発生のメカニズム）

図２０から図２２を使ってイベント発生のメカニズムについて説明する。

【００６０】

図２０はタイムイベントの例である。

前述したとおり、タイムイベントはプレイリスト情報（”ＸＸＸ．ＰＬ”）のイベントリスト（ＥｖｅｎｔＬｉｓｔ）で定義される。タイムイベントとして定義されているイベント、即ちイベントタイプ（Ｔｙｐｅ）が”ＴｉｍｅＥｖｅｎｔ”の場合、イベント生成時刻（”ｔ１”）になった時点で、ＩＤ”Ｅｘ１”を持つタイムイベントがシナリオプロセッサからプログラムプロセッサに対してあげられる。プログラムプロセッサは、イベントＩＤ”Ｅｘ１”を持つイベントハンドラを探し、対象のイベントハンドラを実行処理す

る。例えば、本実施例の場合では、２つのボタンイメージの描画を行うなどを行うことができる。

【００６１】

図２１はメニュー操作を行うユーザーイベントの例である。

前述したとおり、メニュー操作を行うユーザイベントもプレイリスト情報（”ＸＸＸ．ＰＬ”）のイベントリスト（ＥｖｅｎｔＬｉｓｔ）で定義される。ユーザイベントとして定義されるイベント、即ちイベントタイプ（Ｔｙｐｅ）が”ＵｓｅｒＥｖｅｎｔ”の場合、イベント生成時刻（”ｔ１”）になった時点で、当該ユーザイベントがレディとなる。この時、イベント自身は未だ生成されてはいない。当該イベントは、有効規格情報（Ｄｕｒａｔｉｏｎ）で記される期間レディ状態にある。

【００６２】

図２１に描くように、ユーザがリモコンキーの「上」「下」「左」「右」キーまたは「決定」キーを押した場合、先ずＵＯＰイベントがＵＯＰマネージャによって生成されプログラムプロセッサに上げられる。プログラムプロセッサは、シナリオプロセッサに対してＵＯＰイベントを流し、シナリオプロセッサはＵＯＰイベントを受け取った時刻に有効なユーザイベントが存在するかを検索し、対象となるユーザイベントがあった場合は、ユーザイベントを生成し、プログラムプロセッサに持ち上げる。プログラムプロセッサでは、イベントＩＤ”Ｅｖ１”を持つイベントハンドラを探し、対象のイベントハンドラを実行処理する。例えば、本実施例の場合では、プレイリスト＃２の再生を開始する。

【００６３】

生成されるユーザイベントには、どのリモコンキーがユーザによって押されたかの情報は含まれていない。選択されたリモコンキーの情報は、ＵＯＰイベントによってプログラムプロセッサに伝えられ、仮想プレーヤが持つレジスタＳＰＲＭ（８）に記録保持される。イベントハンドラのプログラムは、このレジスタの値を調べ分岐処理を実行することが可能である。

図２２はグローバルイベントの例である。

【００６４】

前述したとおり、グローバルイベントはＢＤディスク全体に関する情報（”ＢＤ．ＩＮＦＯ”）のイベントリスト（ＥｖｅｎｔＬｉｓｔ）で定義される。グローバルイベントとして定義されるイベント、即ちイベントタイプ（Ｔｙｐｅ）が”ＧｌｏｂａｌＥｖｅｎｔ”の場合、ユーザのリモコンキー操作があった場合にのみイベントが生成される。

ユーザが”メニュー”を押した場合、先ずＵＯＰイベントがＵＯＰマネージャによって生成されプログラムプロセッサに上げられる。プログラムプロセッサは、シナリオプロセッサに対してＵＯＰイベントを流し、シナリオプロセッサは、該当するグローバルイベントを生成し、プログラムプロセッサに送る。プログラムプロセッサでは、イベントＩＤ”ｍｅｎｕ”を持つイベントハンドラを探し、対象のイベントハンドラを実行処理する。例えば、本実施例の場合ではプレイリスト＃３の再生を開始している。

【００６５】

本実施例では、単に”メニュー”キーと呼んでいるが、ＤＶＤのように複数のメニューキーがあってもよい。各メニューキーに対応するＩＤを夫々定義することで対応することが可能である。

（仮想プレーヤマシン）

図２３を用いてプログラムプロセッサの機能構成を説明する。

【００６６】

プログラムプロセッサは、内部に仮想プレーヤマシンを持つ処理モジュールである。仮想プレーヤマシンはＢＤ－ＲＯＭとして定義された機能モデルであって、各ＢＤ－ＲＯＭプレーヤの実装には依存しないものである。即ち、どのＢＤ－ＲＯＭプレーヤにおいても同様の機能を実行するできることを保証している。

仮想プレーヤマシンは大きく２つの機能を持っている。プログラミング関数とプレーヤ変数（レジスタ）である。プログラミング関数は、Ｊａｖａ（登録商標）Ｓｃｒｉｐｔをベースとして、以下に記す２つの機能をＢＤ－ＲＯＭ固有関数として定義している。

リンク関数：現在の再生を停止し、指定するプレイリスト、セル、時刻からの再生を開始する
Ｌｉｎｋ（ＰＬ#，Ｃｅｌｌ#，ｔｉｍｅ）
ＰＬ#　：　プレイリスト名
Ｃｅｌｌ#　：　セル番号
ｔｉｍｅ　：　セル内での再生開始時刻

ＰＮＧ描画関数：指定ＰＮＧデータをイメージプレーンに描画する
Ｄｒａｗ（Ｆｉｌｅ，Ｘ，Ｙ）
Ｆｉｌｅ　：　ＰＮＧファイル名
Ｘ　：　Ｘ座標位置
Ｙ　：　Ｙ座標位置

イメージプレーンクリア関数：イメージプレーンの指定領域をクリアする
Ｃｌｅａｒ（Ｘ，Ｙ，Ｗ，Ｈ）
Ｘ　：　Ｘ座標位置
Ｙ　：　Ｙ座標位置
Ｗ　：　Ｘ方向幅
Ｈ　：　Ｙ方向幅

プレーヤ変数は、プレーヤの状態を示すシステムパラメータ（ＳＰＲＭ）と一般用途として使用可能なゼネラルパラメータ（ＧＰＲＭ）とがある。

図２４はシステムパラメータ（ＳＰＲＭ）の一覧である。

ＳＰＲＭ（０）	：	言語コード
ＳＰＲＭ（１）	：	音声ストリーム番号
ＳＰＲＭ（２）	：	字幕ストリーム番号
ＳＰＲＭ（３）	：	アングル番号
ＳＰＲＭ（４）	：	タイトル番号
ＳＰＲＭ（５）	：	チャプタ番号
ＳＰＲＭ（６）	：	プログラム番号
ＳＰＲＭ（７）	：	セル番号
ＳＰＲＭ（８）	：	選択キー情報
ＳＰＲＭ（９）	：	ナビゲーションタイマー
ＳＰＲＭ（１０）	：	再生時刻情報
ＳＰＲＭ（１１）	：	カラオケ用ミキシングモード
ＳＰＲＭ（１２）	：	パレンタル用国情報
ＳＰＲＭ（１３）	：	パレンタルレベル
ＳＰＲＭ（１４）	：	プレーヤ設定値（ビデオ）
ＳＰＲＭ（１５）	：	プレーヤ設定値（オーディオ）
ＳＰＲＭ（１６）	：	音声ストリーム用言語コード
ＳＰＲＭ（１７）	：	音声ストリーム用言語コード（拡張）
ＳＰＲＭ（１８）	：	字幕ストリーム用言語コード
ＳＰＲＭ（１９）	：	字幕ストリーム用言語コード（拡張）

ＳＰＲＭ（２０）　：　プレーヤリージョンコード
ＳＰＲＭ（２１）　：　予備
ＳＰＲＭ（２２）　：　予備
ＳＰＲＭ（２３）　：　再生状態
ＳＰＲＭ（２４）　：　予備
ＳＰＲＭ（２５）　：　予備
ＳＰＲＭ（２６）　：　予備
ＳＰＲＭ（２７）　：　予備
ＳＰＲＭ（２８）　：　予備
ＳＰＲＭ（２９）　：　予備
ＳＰＲＭ（３０）　：　予備
ＳＰＲＭ（３１）　：　予備

なお、本実施例では、仮想プレーヤのプログラミング関数をＪａｖａ（登録商標）Ｓｃｒｉｐｔベースとしたが、Ｊａｖａ（登録商標）Ｓｃｒｉｐｔではなく、ＵＮＩＸ（登録商標）　ＯＳなどで使われているＢ－Ｓｈｅｌｌや、Ｐｅｒｌ　Ｓｃｒｉｐｔなど他のプログラミング関数であっても構わなく、言い換えれば、本発明はＪａｖａ（登録商標）Ｓｃｒｉｐｔに限定されるものでは無い。

（プログラムの例）
図２５及び図２６は、イベントハンドラでのプログラムの例である。
【００６７】
図２５は、２つの選択ボタンを持ったメニューの例である。
セル（ＰｌａｙＬｉｓｔ＃１．Ｃｅｌｌ＃１）先頭でタイムイベントを使って図２５左側のプログラムが実行される。ここでは、最初にゼネラルパラメータの一つＧＰＲＭ（０）に”１”がセットされている。ＧＰＲＭ（０）は、当該プログラムの中で、選択されているボタンを識別するのに使っている。最初の状態では、左側に配置するボタン１が選択されている事を初期値として持たされている。
【００６８】
次に、ＰＮＧの描画を描画関数であるＤｒａｗを使ってボタン１、ボタン２夫々について行っている。ボタン１は、座標（１０、２００）を起点（左端）としてＰＮＧイメージ”１ｂｌａｃｋ．ｐｎｇ”を描画している。ボタン２は、座標（３３０，２００）を起点（左端）としてＰＮＧイメージ”２ｗｈｉｔｅ．ｐｎｇ”を描画している。
また、本セル最後ではタイムイベントを使って図２５右側のプログラムが実行される。ここでは、Ｌｉｎｋ関数を使って当該セルの先頭から再度再生するように指定している。
【００６９】
図２６は、メニュー選択のユーザイベントのイベントハンドラの例である。
「左」キー、「右」キー、「決定」キー何れかのリモコンキーが押された場合夫々に対応するプログラムがイベントハンドラに書かれている。ユーザがリモコンキーを押した場合、図２１で説明したとおり、ユーザイベントが生成され、図２６のイベントハンドラが起動されることになる。本イベントハンドラでは、選択ボタンを識別しているＧＰＲＭ（０）の値と、選択されたリモコンキーを識別するＳＰＲＭ（８）を使って分岐処理を行っている。

条件１）ボタン１が選択されている、かつ、選択キーが「右」キーの場合
ＧＰＲＭ（０）を２に再設定して、選択状態にあるボタンを右ボタン２に変更する。
【００７０】
ボタン１、ボタン２のイメージを夫々書き換える。

条件２）選択キーが「決定（ＯＫ）」の場合で、ボタン１が選択されている場合、プレ

イリスト＃２の再生を開始する

条件３）選択キーが「決定（ＯＫ）」の場合で、ボタン２が選択されている場合、プレイリスト＃３の再生を開始する

上記のようにして実行処理が行われる。

（プレーヤ処理フロー）
次に図２７から図３０を用いてプレーヤでの処理フローを説明する。
【００７１】
図２７は、ＡＶ再生までの基本処理フローである。
ＢＤディスクを挿入すると（Ｓ１０１）、ＢＤ－ＲＯＭプレーヤはＢＤ．ＩＮＦＯファイルの読み込みと解析（Ｓ１０２）、ＢＤ．ＰＲＯＧの読み込み（Ｓ１０３）を実行する。ＢＤ．ＩＮＦＯ及びＢＤ．ＰＲＯＧは共に管理情報記録メモリに一旦格納され、シナリオプロセッサによって解析される。
【００７２】
続いて、シナリオプロセッサは、ＢＤ．ＩＮＦＯファイル内のファーストイベント（ＦｉｒｓｔＥｖｅｎｔ）情報に従い、最初のイベントを生成する（Ｓ１０４）。生成されたファーストイベントは、プログラムプロセッサで受け取られ、当該イベントに対応するイベントハンドラを実行処理する（Ｓ１０５）。
ファーストイベントに対応するイベントハンドラには、最初に再生するべきプレイリスト情報が記録されていることが期待される。仮に、プレイリスト再生が指示されていない場合には、プレーヤは何も再生することなく、ユーザイベントを受け付けるのを待ち続けるだけになる。この場合、ユーザイベントを受け付けるのを待ち続けることになる（Ｓ２０１）。ＢＤ－ＲＯＭプレーヤはがユーザからのリモコン操作を受け付けると、ＵＯＰマネージャはプログラムマネージャに対してＵＯＰイベントを立ち上げる（Ｓ２０２）。
【００７３】
プログラムマネージャは、ＵＯＰイベントがメニューキーによるものであるかを判別し（Ｓ２０３）、メニューキーの場合は、シナリオプロセッサにＵＯＰイベントを流し、シナリオプロセッサがユーザイベントを生成する（Ｓ２０４）。プログラムプロセッサは生成されたユーザイベントに対応するイベントハンドラを実行処理する（Ｓ２０５）。
図２８は、ＰＬ再生開始からＶＯＢ再生開始までの処理フローである。
【００７４】
前述したように、ファーストイベントハンドラまたはグローバルイベントハンドラによってプレイリスト再生が開始される（Ｓ３０１）。シナリオプロセッサは、再生対象のプレイリスト再生に必要な情報として、プレイリスト情報”ＸＸＸ．ＰＬ”の読み込みと解析（Ｓ３０２）、プレイリストに対応するプログラム情報”ＸＸＸ．ＰＲＯＧ”の読み込みを行う（Ｓ３０３）。続いてシナリオプロセッサは、プレイリストに登録されているセル情報に基づいてセルの再生を開始する（Ｓ３０４）。セル再生は、シナリオプロセッサからプレゼンテーションコントローラに対して要求が出さる事を意味し、プレゼンテーションコントローラはＡＶ再生を開始する（Ｓ３０５）。
【００７５】
ＡＶ再生の開始（Ｓ４０１）を開始すると、プレゼンテーションコントローラは再生するセルに対応するＶＯＢの情報ファイル（ＸＸＸ．ＶＯＢＩ）を読み込み及び解析をする（Ｓ４０２）。プレゼンテーションコントローラは、タイムマップを使って再生開始するＶＯＢＵとそのアドレスを特定し、ドライブコントローラに読み出しアドレスを指示し、ドライブコントローラは対象となるＶＯＢデータを読み出し（Ｓ４０３）、ＶＯＢデータがデコーダに送られ再生が開始される（Ｓ４０４）。ＶＯＢ再生は、当該ＶＯＢの再生区間が終了するまで続けられ（Ｓ４０５）、終了すると次のセル再生Ｓ３０４へ移行する。次にセルが無い場合は、再生が停止する（Ｓ４０６）。

【００７６】
図２９は、ＡＶ再生開始後からのイベント処理フローである。
ＢＤ－ＲＯＭプレーヤはイベントドリブン型のプレーヤモデルである。プレイリストの再生を開始すると、タイムイベント系、ユーザイベント系、字幕表示系のイベント処理プロセスが夫々起動され、平行してイベント処理を実行するようになる。
Ｓ５００系の処理は、タイムイベント系の処理フローである。
【００７７】
プレイリスト再生開始後（Ｓ５０１）、プレイリスト再生が終了しているかを確認するステップ（Ｓ５０２）を経て、シナリオプロセッサは、タイムイベント発生時刻になったかを確認する（Ｓ５０３）。タイムイベント発生時刻になっている場合には、シナリオプロセッサはタイムイベントを生成し（Ｓ５０４）、プログラムプロセッサがタイムイベントを受け取りイベントハンドラを実行処理する（Ｓ５０５）。
【００７８】
ステップＳ５０３でタイムイベント発生時刻になっていない場合、または、ステップＳ５０４でイベントハンドラ実行処理後は再度ステップＳ５０２へ戻り、上述した処理を繰り返す。また、ステップＳ５０２でプレイリスト再生が終了したことが確認されると、タイムイベント系の処理は強制的に終了する。
Ｓ６００系の処理は、ユーザイベント系の処理フローである。
【００７９】
プレイリスト再生開始後（Ｓ６０１）、プレイリスト再生終了確認ステップ（Ｓ６０２）を経て、ＵＯＰ受付確認ステップの処理に移る（Ｓ６０３）。ＵＯＰの受付があった場合、ＵＯＰマネージャはＵＯＰイベントを生成し（Ｓ６０４）、ＵＯＰイベントを受け取ったプログラムプロセッサはＵＯＰイベントがメニューコールであるかを確認し（Ｓ６０５）、メニューコールであった場合は、プログラムプロセッサはシナリオプロセッサにイベントを生成させ（Ｓ６０７）、プログラムプロセッサはイベントハンドラを実行処理する（Ｓ６０８）。
【００８０】
ステップＳ６０５でＵＯＰイベントがメニューコールで無いと判断された場合、ＵＯＰイベントはカーソルキーまたは「決定」キーによるイベントである事を示している。この場合、現在時刻がユーザイベント有効期間内であるかをシナリオプロセッサが判断し（Ｓ６０６）、有効期間内である場合には、シナリオプロセッサがユーザイベントを生成し（Ｓ６０７）、プログラムプロセッサが対象のイベントハンドラを実行処理する（Ｓ６０８）。
【００８１】
ステップＳ６０３でＵＯＰ受付が無い場合、ステップＳ６０６で現在時刻がユーザイベント有効期間に無い場合、または、ステップＳ６０８でイベントハンドラ実行処理後は再度ステップＳ６０２へ戻り、上述した処理を繰り返す。また、ステップＳ６０２でプレイリスト再生が終了したことが確認されると、ユーザイベント系の処理は強制的に終了する。
【００８２】
図３０は字幕処理のフローである。
プレイリスト再生開始後（Ｓ７０１）、プレイリスト再生終了確認ステップ（Ｓ７０２）を経て、字幕描画開始時刻確認ステップに移る（Ｓ７０３）。字幕描画開始時刻の場合、シナリオプロセッサはプレゼンテーションコントローラに字幕描画を指示し、プレゼンテーションコントローラはイメージプロセッサに字幕描画を指示する（Ｓ７０４）。ステップＳ７０３で字幕描画開始時刻で無いと判断された場合、字幕表示終了時刻であるかを確認する（Ｓ７０５）。字幕表示終了時刻であると判断された場合は、プレゼンテーションコントローラがイメージプロセッサに字幕消去指示を行い、描画されている字幕をイメージプレーンから消去する（Ｓ７０６）。
【００８３】

字幕描画ステップＳ７０４終了後、字幕消去ステップＳ７０６終了後、または、字幕表示終了時刻確認ステップＳ７０５で当該時刻でないことが判断された場合、ステップＳ７０２に戻り、上述した処理を繰り返す。また、ステップＳ７０２でプレイリスト再生が終了したことが確認されると、字幕表示系の処理は強制的に終了する。

（実施例２）

次に本発明の第２の実施の形態について説明する。

【００８４】

第２の実施の形態は、ＢＤアプリケーションにおいてより豊かなインタラクティブ性を実現するため、Ｊａｖａ（登録商標）のようなプログラミング環境を導入することに関する内容である。基本的には第１の実施例に基づく内容であり、拡張または異なる部分を中心に説明する。

（プログラミング環境の導入）

実施例１において、図４が示す「ＢＤ再生プログラミング」の主な目的は、ＡＶデータの再生順序を規定する、あるいは、ユーザーの操作によりＡＶの再生順序を変更することであった。しかし、ＢＤ再生プログラミングとしてＪａｖａ（登録商標）のような汎用的なプログラミング環境を利用するのであれば、ＡＶ再生の制御のみならず様々なアプリケーションを提供することが可能である。本実施例では、汎用的なプログラミング環境としてＪａｖａ（登録商標）を想定しているが、Ｃ言語やその他のプログラミング言語であっても、同様である。

【００８５】

ＡＶ再生以外のアプリケーションが利用可能になったとしても、あるカテゴリーごとにＡＶ再生やアプリケーションを認識する方が、ユーザーには分かりやすくなる。カテゴリーとは映画本編や本編以外の付属のコンテンツとして提供される各種ゲームといったカテゴリーであり、カテゴリーをユーザーに認識できるようにするため、タイトルという単位を用意することとする。

【００８６】

実施例１においては、図１８が示すとおり、タイトルとは再生順序が記述されているプレイリストへのリンク情報であり、タイトルが指定されると関連付けられたプレイリストを参照して、プレイリストが参照しているセルやＡＶストリームが再生された。

本実施例においてタイトルが指定されると、プログラムを介したＡＶストリームが再生される場合もあれば、ＡＶストリームを再生しないアプリケーションが起動する場合もある。

【００８７】

図３１は図６を本実施例に合わせて拡張したものである。ディスクより読み込まれたプログラム（３１０１）は、プログラム処理部（３１０２）で処理される。プログラムは管理情報処理部やプレゼンテーション処理部に指示を出し、プレイリストを介してＡＶストリームを再生するものもあれば、通常のＪａｖａ（登録商標）アプリケーション同様にイメージプレーン（３１０４）に対して描画指示（３１０５）を行うものもある。

【００８８】

ＡＶストリームの再生のみを指示するアプリケーションの例としては、本編映画再生をユーザーの指示に従って行う単純な再生アプリケーションが挙げられる。ＡＶ再生を行わないアプリケーションの例としては、フル画面をプログラムで描画するゲームのようなアプリケーションが挙げられる。ＡＶを再生しながらアプリケーションとして描画することも可能であり、再生中のＡＶの上にプログラムが描画したイメージ図を重ねたり、移動させてアニメーションを行ったりすることも可能である。

【００８９】

タイトルはユーザーに認識されるＡＶストリームの再生及びアプリケーションの実行の単位であるが、同時にプレーヤがＡＶストリームの再生及びアプリケーションの実行を管

理する単位でもある。

（アプリケーションの管理）

図３２はタイトルリスト（３２０１）が、アプリケーションの管理単位としてのタイトルを指し示している様子を示している。

【００９０】

タイトル＃１は映画本編が再生されるタイトルである。タイトル＃１が選択されると、本編映像再生用のアプリケーション（３２０２）が起動され、本編映像の再生が開始される。本編再生用のアプリケーションは、リモコンキーによる早送り・スキップなどのイベントを受けて、再生制御部に指示を出しプレーヤの動作をコントロールする。タイトル＃１が起動されると本編再生用アプリケーションとともに、ポップアップメニューを出現させるためのアプリケーション（３２０３）や、本編再生中の映像に出てきた品物を購入できるようなショッピングカートアプリケーション（３２０４）なども起動させる。

【００９１】

本編再生から別の目的のタイトルに移りたい場合は、タイトル切替を行う。たとえば本編再生中に出てきた品物を購入するためにオンラインショッピングアプリケーションを起動させたければ、オンラインショッピングを行うタイトル＃２に移動する。

タイトル＃１からタイトル＃２に移動すると、タイトル＃１の場合同様にタイトル＃２に関連したアプリケーション（３２０５）が起動されるとともに、タイトル＃２に関係ないアプリケーション（３２０２、３２０３）は停止する。タイトル＃１にもタイトル＃２にも関係するアプリケーション（３２０４）は、どちらのタイトルでも起動し続けることができ、タイトル＃１とタイトル＃２の間のタイトル遷移中に停止と再起動が起こるわけではない。

【００９２】

タイトル＃３はタイトル＃１及びタイトル＃２とは独立したタイトルであり、タイトル＃３が選択されると、タイトル＃１及びタイトル＃２で起動されていたアプリは停止し、タイトル＃３に関連したアプリケーション（３２０６）が起動される。

図３３はこれまで述べたような、タイトル単位でのアプリケーションの管理を実現するための管理情報について示している。

【００９３】

アプリケーション管理情報（３３０１）は、タイトル番号（３３０２）とアプリケーションＩＤ（３３０３）のペアを管理しており、指定されたタイトル中でどのアプリケーションが起動されるかを管理している。図３３中の例では、タイトル＃１において、アプリケーション＃１とアプリケーション＃２が起動され、タイトル＃２においてアプリケーション＃３が起動される。

【００９４】

タイトルが選択されるとすぐにアプリケーション管理情報にしたがって願連するアプリケーションが起動され（３３０４）、タイトルが変わると次のタイトルで関連付けられていないアプリケーションは終了し（３３０５）、次のタイトルに関連付けられているアプリケーションが起動される（３３０６）。

なお、アプリケーションＩＤとしては、Ａｐｐｌｅｔやｘｌｅｔなどの起動インターフェースを持ったクラスファイル名などでよい。

図３４は、拡張することによりタイトル開始時の起動状態を設定可能にしたアプリケーション管理情報について示している。

【００９５】

図３３で示されたアプリケーション管理情報は、あるタイトルが開始されたときには必ず関連付けられたアプリケーションが起動された。そこでタイトルが開始されたときに起動されるアプリケーションと起動されないアプリケーションを区別するために、タイトル開始時の起動状態（３４０１）を持たせることにする。

アプリケーション＃１はタイトル＃１に開始時に自動的に起動されるアプリケーションである。アプリケーション＃２は同じくタイトル＃１に関連付けられているが、タイトル開始時に自動的に起動はされない。アプリケーション＃２は、他のアプリケーションによって起動される（３４０２）、あるいはユーザー操作などのイベントにより起動される。

【００９６】

アプリケーション＃２は、タイトル＃１開始時に自動的に起動されることはないが、タイトル＃２に切り替わるときに起動していて、かつ、タイトル＃２に登録されていなければ、自動的に終了する。

アプリケーションの終了の仕方としては、アプリケーション＃２のように他のアプリやイベントによって起動する場合と同様、他のアプリやイベントによって終了することもある（３４０４）。このような場合、起動属性がＡｕｔｏＲｕｎ（自動的に起動）であっても、タイトル中で自動的に再起動することはない。タイトルが切り替わり、再びタイトル＃１が開始されたときは、自動的に起動される。

【００９７】

図３５は、アプリケーション管理情報の起動状態の拡張を行い、複数タイトル間で継続的にアプリケーションを起動しておく方法、アプリケーションのサスペンド状態を実現する方法を説明するためのものである。

複数のタイトルに対して、同じアプリケーションを登録しておくことにより、タイトルを越えてアプリケーションを起動させ続けることが可能となる。アプリケーション管理情報において、アプリケーション＃１はタイトル＃１にもタイトル＃２にも登録（３５０１、３５０２）されている。アプリケーション＃１はタイトル＃１開始時にＡｕｔｏＲｕｎ属性により自動的に起動される。タイトル＃１からタイトル＃２に遷移する際、アプリケーション＃１はタイトル＃２にも登録されているため、タイトル＃１終端で終了することなく、タイトル＃２に遷移する。このように複数タイトルに同一アプリケーションを登録しておけば、登録されているタイトル間を遷移する際にアプリケーションが終了後再起動されるようなことはない。

【００９８】

複数タイトルをまたげるアプリケーションであったとしても、登録されていないタイトルに遷移する際は終了される。アプリケーション＃１の例では、タイトル＃２からタイトル＃３に遷移する際は、アプリケーション＃１は終了される。　次に、アプリケーションをサスペンド状態に遷移させる方法を説明する。通常の起動状態では、Ｊａｖａ（登録商標）ＶＭなどの実行環境において、実行環境内のメモリを割り当てられ、ＣＰＵリソースも割り当てられることにより実行されている。起動していないアプリケーションは、メモリもＣＰＵリソースも割り当てられていない。それに対して、サスペンド状態とは、実行環境内のメモリに格納されているが、ＣＰＵリソースは割り当てられていない状態であり、ＣＰＵリソースを割り当てられれば、いつでも実行可能となる。このようなサスペンド状態を管理するために、起動属性として「サスペンド」を追加する。起動属性としてサスペンドを指定されているアプリケーションは、起動していればサスペンド状態に遷移する。

【００９９】

図３５においてアプリケーション＃２がサスペンドされるアプリケーションである。アプリケーション＃２はタイトル＃１において起動され、タイトル＃２に遷移する際に自動的にサスペンド状態となる。アプリケーション＃２はタイトル＃２中において、実行環境のメモリ内には存在できるがＣＰＵリソースは割り当てられない状態である。タイトル＃３のアプリケーション管理情報では、アプリケーション＃２は通常状態（サスペンドではない）の起動属性を持っているため、タイトル＃２からタイトル＃３に遷移する際にアプリケーション＃２はレジュームされ、再び起動状態に戻る。

【０１００】

図３６は、起動属性によって、タイトル間遷移時に状態がどのように変化するかを示した図表である。これまでの例として述べてきたように、アプリケーションが起動していな

い状態で起動属性としてＡｕｔｏＲｕｎが指定されているタイトルに遷移すると、アプリケーションは自動的に起動される。すでにアプリケーションが起動された状態で起動属性がサスペンドであるタイトルに遷移すると、アプリケーションはサスペンドされる。アプリケーションがサスペンドされた状態で、起動属性がＡｕｔｏＲｕｎあるいは指定されていない（サスペンドではない）タイトルに遷移すると、アプリケーションはレジュームされ、起動状態に戻る。それ以外の場合はアプリケーションの状態に変化は起こらず、前の状態をそのまま継続する。

【０１０１】

なお、アプリケーションは他のアプリケーションからの制御やイベントを受けてアプリケーション自体が反応することによって、アプリケーションの状態を変更することもある。図３４のアプリケーション＃２は、他のアプリケーションからの起動される例である。

なお、アプリケーションがサスペンドされている状態で、起動属性が指定されてないタイトルに遷移した際、レジュームするのではなく何もしなくてもよい。　図３７はアプリケーション管理情報の管理単位の区分を示している。各単位ごとの管理情報は、単位ごと（タイトルの場合はタイトルごと、図３７の例）に管理情報を分割して別々に記録（３７０１）してもよいし、タイトルごとの管理情報でもまとめて図３５のアプリケーション管理情報ように管理してもよい。

【０１０２】

ここまでのアプリケーション管理情報は、カテゴリの単位としてのタイトルを基本とし、タイトル単位でアプリケーションの起動や終了を管理していた。アプリケーションによっては複数のタイトルをまたげるものもあるが、ディスクに存在する全てのタイトルで有効なアプリケーションの場合は、全ての管理テーブルに書かなければならず、煩雑になる。そこで、ディスク全体で有効なアプリケーションに関しては、ディスク単位の管理情報（３７０２）を用意すれば、ディスクで一括して管理できるため管理が容易になる。

【０１０３】

同様に、ディスク３枚組のようなパッケージがある場合は、ディスクセット単位でアプリケーションを管理するアプリケーション管理情報（３７０３）を、どのディスクにも同じ情報を記録しておくことで実現可能である。

また、これらの階層別のアプリケーション管理情報を組み合わせて使うことも可能である。組み合わせて利用する場合で、どちらか一方にしか登録されていない場合（たとえばディスクのアプリケーション管理情報には登録されていて、タイトルのアプリケーション管理情報には登録されていない場合）は、管理テーブルの和集合をとってプレーヤ内で独自のアプリケーション管理情報を構成すればよい。２つの管理テーブルで情報が異なる場合は、より有効範囲が狭い管理テーブル（先ほどの例では、タイトルのアプリケーション管理情報）を優先するようにしてもよい。

【０１０４】

複数のディスクにまたがっても、アプリケーションを一意に指定するため、アプリケーションＩＤとして、オーサーが決めた独自の番号だけではなく、オーサーごとに割り振られた一意なＩＤとオーサーが決めた独自の番号を組み合わせることにより、他社製のコンテンツとアプリケーションＩＤが重なることを防ぐことも可能である。

なお、複数ディスクにまたがる管理情報の場合、ディスクが抜かれた状態でのアプリケーションの動作を規定することも可能である。ディスクが抜かれた状態ではアプリケーションを一時停止状態にする、あるいは、使用できる機能を限定しなければ、悪意を持ったアプリケーションやバグにより、ディスク交換時あるいはディスクを抜いたあともアプリケーションが起動し続け、ユーザーが気づかないうちにデータの改変あるいはネットワークへのアクセスを行うかもしれない。

【０１０５】

図３８はタイトルより短い単位でアプリケーションを管理する方法を示している。ＡＶストリームが再生されるタイトルの場合、タイトル再生中にいくつかのチャプター（３８０１）が用意されていることがある。チャプターはＡＶストリームの再生時間軸上に順番

に並んでいるため、時間軸上の特定の区間でのみ有効なアプリケーションを規定したい場合、タイトル指定とともにさらに時間の限定を加えてチャプターあるいはチャプターの範囲を指定する（３８０２）ことにより、アプリケーションの有効期間をタイトル中の時間軸上の特定期間に限定する（３８０３）ことができる。

【０１０６】

なお、さらに細かい単位でアプリケーションを管理したい場合には、タイトル内のストリームの時間軸を利用し、再生開始時点よりＡ秒後からＢ秒後のように時間指定することも可能である。

（アプリケーションマネージャ）

アプリケーション管理情報を利用して、アプリケーションを起動及び終了させる仕組みについて述べる。

【０１０７】

図３９はプログラミング環境を用いて再生を制御する部分の大まかなブロック図である。ディスク（３９０１）から読み込まれたプログラムやプログラムから使用するイメージデータは、ローカルメモリ（３９０２）に蓄積される。ローカルメモリは図６のプログラム記憶メモリと同等であるが、プログラムだけではなくイメージデータなども格納される。ローカルメモリ上のプログラム及びデータは、Ｊａｖａ（登録商標）仮想マシン（３９０３）などのプログラム処理部が必要とした時に、プログラム処理部のワークメモリ（３９０４）内に読み込まれる。ワークメモリはプログラムを実行する際にプログラム処理部が利用するメモリであり、Ｊａｖａ（登録商標）仮想マシンではヒープ領域あるいはヒープメモリなどと呼ばれるメモリである。ワークメモリに読み込まれプログラム処理部によりプログラムが実行されるとアプリケーション（３９０５）が起動される。図６におけるプログラム処理部は、Ｊａｖａ（登録商標）仮想マシンとワークメモリが含まれている。アプリケーションの動作によっては、プログラム処理部を介して、ＡＶストリーム再生部（３９０６）に指示を出すことも可能である。

【０１０８】

アプリケーション管理情報は、アプリケーションマネージャ（３９０７）によって読み込まれ、アプリケーションマネージャがプログラム処理部を制御することのより、アプリケーションの起動や終了を行う。タイトル遷移時に、アプリケーションマネージャはアプリケーション管理情報を参照し、該当するアプリケーションのプログラムをローカルメモリより読み込んで実行、あるいは必要ないアプリケーションを終了する。

【０１０９】

なお、ディスクからローカルメモリにデータが読み込まれる時、あるいは、ローカルメモリからワークメモリにデータが読み込まれるときに、プログラムやイメージデータのサイズをチェック、プログラムのコードにデータ落ちなどが存在しないか、プログラムを動作させるＪａｖａ（登録商標）仮想マシンとバージョンが合っているかなどチェックする機能があってもよい。また、不正なプログラムが動作することを防ぐためプログラムを暗号化してディスクに記録しておき、読み込む際に複合化するなどセキュリティを確保する機能があってもよい。

図４０はアプリケーションマネージャによるアプリケーションの起動及び終了の様子を示している。

【０１１０】

タイトル管理情報を参照してタイトル開始時にアプリケーションマネージャがアプリケーションの起動する（４００１）以外に、ユーザー操作により発生するイベントや時間イベントに応じて、アプリケーションマネージャがアプリケーションを起動（４００２）してもよい。また、他のアプリケーションからの制御によりアプリケーションを起動する（４００３）ことも可能である。この場合、アプリケーションが直接他のアプリを起動してもよいし、アプリケーションマネージャに指定のアプリケーションを起動するように問い

合わせてもよい。

【０１１１】

アプリケーションの終了は、タイトル開始時と同様、タイトル終了時にアプリケーションマネージャが次のタイトルで登録されていないアプリケーションＩＤを持つアプリを終了（４００４）させる、あるいは、ユーザー操作により発生するイベントや時間イベントに応じてアプリケーションを終了（４００５）させてもよい。他のアプリケーションから終了（４００６）してもよい。この場合、直接アプリケーションを終了させてもよいし、アプリケーションマネージャに、指定のアプリケーションを終了するように問い合わせてもよい。

【０１１２】

図４１はアプリケーションマネージャによるアプリケーションの終了方法について示している。アプリケーションを終了させる際は、アプリケーションマネージャより終了させたいアプリケーションに対して終了イベントを通知し、アプリケーションのイベント受信部（４１０１）がイベントを受け、アプリケーション内の終了プロセス（４１０２）を実行することによりアプリケーションが自ら終了する。アプリケーションが終了イベント受信部と対応した終了プロセスを持たない（４１０３）ため自ら終了しない、あるいは、終了プロセスが間違っておりアプリケーションが自ら終了しない場合などは、一定時間経過後アプリケーションマネージャがアプリケーションを強制的に終了させる。

【０１１３】

この方法はサスペンド時やレジューム時にも応用可能であり、アプリケーションマネージャがアプリケーションを強制的にサスペンドあるいはレジュームする前に、アプリケーション独自のプロセスを処理する仕組みを提供することが可能である。

また、アプリケーションマネージャは、アプリケーション自体に終了やサスペンドの処理を任せるのだけではなく、アプリケーションが終了やサスペンドなどのイベントに正常に反応しなかった場合、強制的に処理することにより、個々のアプリケーションだけではなく全体の動作を管理することが可能となる。この仕組みにより、たとえあるタイトル内でアプリケーションの動作がおかしくなったとしても、タイトルを切り替えることによりアプリケーションマネージャにより登録されていないアプリケーションは終了され、アプリケーションの管理を保証する仕組みを提供することが可能となる。全てのアプリケーションを終了させるために、どのアプリケーションも実行されないタイトルを用意しておくことにより、そのタイトルに遷移することで全てのアプリケーションを停止することも可能である。このようなどのアプリケーションも動作できない、あるいは安全な特別なアプリケーションしか起動できないタイトルは、アプリケーションの動作がおかしくなったときに正常状態に復帰する手段として有効である。映画コンテンツなどの場合は、メニューや再生開始時の注意書きを表示するタイトルなどを利用するとよい。

（メモリ管理）

図４２はプログラムやプログラムが使用するイメージデータなどが蓄積されるローカルメモリと、プログラムが実行されるときにデータを読み込んでおくワークメモリの関係と役割について示している。

【０１１４】

ローカルメモリはディスク上のデータを一時的に蓄えておくキャッシュとしての役割を果たしている。ディスクにはＡＶストリームとプログラムやイメージデータが記録されているが、ＡＶストリームを再生時に起動するアプリケーションのプログラムを読み込んでいては、ディスク上でシークが発生してしまい、最悪ＡＶストリームの再生が一時的に停止してしまうかもしれない。そこで、ＡＶストリームの再生を開始する前や再生が一時的に停止してもよい点において、アプリケーションを起動するために必要なプログラムやデータを読み込み蓄積しておくことにより、ＡＶストリームの再生中はＡＶ再生のため以外のシークを発生させないようにする。

【０１１５】

なお、ハードディスクドライブのようにヘッドのシークやディスクの回転待ちが短時間である場合は、必要になったときにプログラムやデータを読み込み、必要最低限のサイズを持ったローカルメモリにデータを蓄積しておいてもよい。

アプリケーションマネージャがアプリケーションを起動しようとする際、ローカルメモリより必要なプログラムをワークメモリに読み込んでＪａｖａ（登録商標）仮想マシン内で実行する。アプリケーション起動中は、アプリケーションが必要とするプログラム及びデータは、基本的にローカルメモリにすでに読み込まれていることが望ましい。Ｊａｖａ（登録商標）仮想マシンから直接ディスク上のファイルを読み込むことはなく、プログラムやデータは必ずローカルメモリ上に蓄積される。

【０１１６】

アプリケーションが終了すると、Ｊａｖａ（登録商標）仮想マシンはワークメモリから不要なデータを解放（ガーベッジコレクション）する。ただし、アプリケーションが終了したとしてもローカルメモリから自動的にデータが削除されるわけではない。アプリケーションが再び起動するときにはローカルメモリから読み込む必要があるため、アプリケーションが必要なくなるまで、ローカルメモリにデータを保持しておくと、ディスクから再び同じデータを読み込む必要がなくなり効率がよい。

（仮想ドライブ）

ローカルメモリはＪａｖａ（登録商標）仮想マシンから仮想ドライブのようにアクセス可能なメモリである。ディスクより読み込まれたデータはローカルメモリ内でディレクトリ構造などを構成し、アプリケーションからはＲＡＭメモリドライブ上のファイルのようにアクセス可能である。

【０１１７】

図４３はディスク上のファイルとローカルメモリ上のファイルを管理する仕組みを示している。（Ａ）のような管理情報があれば、ディスク上のファイルをローカルメモリ上に読み込んだ際、ローカルメモリ上ではどのディレクトリのなんというファイル名で扱えばよいか、規定することが可能である。また（Ｂ）のように、読み込むべきファイルの一覧作り、ローカルメモリ内でのファイル名は、データとともにヘッダ情報内で管理してもよい。このような仕組みでディスク上のデータは、ローカルメモリ上に配置される。ただし、ファイル数が多くなると、管理が煩雑になってしまう。

【０１１８】

なお、ファイル名の代わりに新たにロケータを定義してもよい。ローカルメモリだけではなく、ハードディスクなどの高速なアクセスが可能なストレージとローカルメモリを統一的に扱えるロケータを定義することにより、少ないメモリで多くのデータを読み込んでおくことが可能である。ハードディスクなどのストレージとローカルメモリの管理をプレーヤの実装にまかせるならば、メモリのページングなどの技術も導入可能である。

【０１１９】

そこで図４４のように、ローカルメモリ上でのファイル名やディレクトリ構造も含めて、複数のファイルをひとまとめにして管理することができる構造を利用することで、ディスク上からは１つのファイルを読み込むだけで、ローカルメモリ上で複数のファイルやディレクトリを構成可能とする。このようにディレクトリ構造を含めて複数のファイルをひとまとめに扱えるデータ構造として、ｔａｒファイルやｚｉｐファイルを利用してもよい。

【０１２０】

なお、ファイルの解放の容易性を考慮し、ディレクトリごとにデータをまとめ、データを解放するときは、ディレクトリごと解放する方にしてもよい。

（データ読み込みのタイミング）

ワークメモリのデータの読み込み及び解放は、アプリケーション管理情報にしたがって行われる。アプリケーションの起動及び終了を指示するトリガーとなるのはアプリケーシ

ョンマネージャであり、実際にデータを読み込んで実行するのはＪａｖａ（登録商標）仮想マシンである。

【０１２１】

それに対し、データがディスクからローカルメモリに読み込まれるタイミングは、データが必要になる期間に依存する。データが必要となる期間は、アプリケーションがデータを必要とする期間であり、タイトルなどのアプリケーションの管理単位と同じ単位で管理される。

図４５はデータの読み込み及びローカルメモリからの解放のタイミングを決める、データ管理情報（４５０１）を示した図である。データ管理情報は、タイトル番号（４５０２）と該当するタイトルで必要となるデータ名あるいはデータを固めて保持しているデータブロック名（４５０３）のペアが管理されている。あるタイトルが選択されると、選択されたタイトルで必要となるデータをローカルメモリに読み込み（４５０４）、タイトル切替によって別のタイトルに遷移する際に、遷移先のタイトルに登録されていないデータはローカルメモリより解放（４５０５）する。

【０１２２】

なお、読み込み及び解放のタイミングは、アプリケーション管理情報と同様に、タイトル単位のほか、ディスク単位やディスクセット単位、あるいはタイトルより小さな単位として、チャプターや時間指定も可能である。

また、有効期間ごとにデータをまとめて、一つのファイルとしてディスクに配置しておくと、データ管理情報が簡潔になる。

（リソースの管理）

アプリケーション管理情報を利用することにより、タイトル中で実行されるアプリケーション数やアプリケーションの実行に必要なワークメモリの容量を有限なサイズに抑えることが可能となる。アプリケーション管理情報がなければ、プログラマの勘違いやプログラムのバグ・誤動作により、アプリケーションが想定数以上に起動してしまい、ワークメモリに入りきらず、全てのアプリケーションの動作が不安定になるかもしれない。ワークメモリなどのリソースを制限するため、プログラムが間違っていたとしても指定されたアプリケーション以外は起動しないように、アプリケーション管理情報及びアプリケーションマネージャによる制御が有効である。

【０１２３】

アプリケーションの起動をアプリケーション自身に任せている場合、プログラムを解析してどのアプリケーションがどの時点で起動しているか判断しなければならず、また、全ての条件を把握しなければ正確に状態をつかめないので、デバッグ及びベリファイすることが非常に困難である。それに対し、アプリケーション管理情報はプログラム内ではなく静的なデータとして保存されているため、どのタイトル中にどのアプリケーションが起動されるあるいは起動される可能性があるか容易に把握でき、各アプリケーションごとに必要となるリソースを見積もることができれば、タイトル中に必要なリソースのトータルサイズも見積もることが可能となる。

【０１２４】

データ管理情報も同様であり、タイトル中にローカルメモリに読み込まれるデータサイズを容易に割り出すことができるため、ローカルメモリのサイズ内に収まるかどうか判断することができ、ベリファイに有効に利用することが可能である。

この管理方法は、ディスクで配布されるコンテンツやアプリケーションが利用するメモリなどのリソースのサイズを常に一定とする方法である。図４６のように、年月が経つにつれてメモリの容量が大容量化され、プレーヤに搭載されたメモリ容量が大きくなったとしても、ディスクは古いプレーヤで搭載されていたメモリ容量で全てのアプリケーションが動作するように作らなければ、そのディスクが全てのプレーヤで動作することが保証されない。あるいは、大容量のメモリをめいいっぱい利用するアプリケーションを記録したディスクがあったとしても、メモリを少ししか搭載されていないプレーヤではそのディス

クに記録されたアプリケーションを実行できなくなってしまう。図中のデータ＃３及びデータ＃４は古いプレーヤではローカルメモリに読み込むことができず、イメージを表示したりプログラムを実行することができない。そのため、使用できるメモリ容量の最大値を制限することは、全てのプレーヤでの動作保証には有効だが、プログラム実行環境の進化においていかれることになりかねない。

【０１２５】

アプリケーション管理情報やデータ管理情報を拡張、リソースの最低必須サイズを規定することにより、メモリが少ないプレーヤでも動作保証され、メモリが多いプレーヤでは余分にデータを読み込むことにより、データ読み込みのレスポンスを良くしてユーザーの操作を快適にしたり、メモリが少ないプレーヤでは実行できないあるいは品質の悪かったアプリケーションが、同じディスクを使ってもメモリの多いプレーヤでは実行できるとあるいは品質が良くなる仕組みを導入する。

【０１２６】

図４６を用いて説明すると、データ＃１やデータ＃２は映画を再生する、あるいは必ず実行されなければならないアプリケーションのデータであり、これらのデータは古いプレーヤでも新しいプレーヤでも必ずローカルメモリ上に読み込まれる。それに対し、データ＃３やデータ＃４のデータは、大容量のメモリが積まれている新しいプレーヤでのみ実行されればよい、おまけ的なプログラムである。

【０１２７】

このような仕組みにより、どのプレーヤでも最低限のコンテンツの再生やアプリケーションの実行を保証でき、メモリサイズの大容量化などリソースの進化に対応できる仕組みを提供することが可能である。

なお、ワークメモリの管理はローカルメモリのように行うのは困難である。ローカルメモリはディスクから読み込まれたデータがそのまま配置されるため、データサイズを容易に見積もることが可能であるが、ワークメモリはＪａｖａ（登録商標）仮想マシンのヒープ領域であり、読み込まれたプログラムのデータ以外にプログラムを管理するデータや変数や配列のために確保したメモリも考慮しなければならず、Ｊａｖａ（登録商標）仮想マシンの実装による影響が大きい。そのため、ワークメモリに関してはメモリのサイズによって動作保証を行うのではなく、アプリケーションが規定サイズのイメージデータを表示可能なことや、規定個数のスレッドを生成できるといった最低限必要な機能を満たすことで保証するのが望ましい。あるいは、ワークメモリの上限値をオーバーしてアプリケーションがメモリを利用しようとしたときに、オーバーフローしていることを通知するイベントあるいは例外処理を発生させ、ワークメモリが不足したときの対策をアプリケーションに行わせる方法も可能である。

（優先度情報を用いたデータ読み込み）

図４７はデータ管理情報の拡張を行い、ローカルメモリサイズに応じて読み込むデータを選択する仕組みについて示している。データ管理情報に有効期間としてのタイトル番号、そのタイトル中で読み込まれるデータに対応させて、データの読み込み優先度情報（４７０１）を追加する。もっとも単純な優先度情報は、図表中の例のように、必ず読み込まなければいけないデータ（ｍａｎｄａｔｏｒｙ）とメモリに余裕があれば読み込む（ｏｐｔｉｏｎａｌ）の２つを区別できる情報であればよい。

【０１２８】

タイトル＃１において、データ＃１とデータ＃２はローカルメモリに必ず読み込まれなければならないデータ（４７０２）である。これらのデータとして、映画の再生行うアプリケーションはメニューを表示するアプリケーションなど、どのプレーヤでも実行されなければならないようなプログラムなどが該当する。読み込み必須のデータの合計サイズは、どのプレーヤでも必ず搭載されていることが保証されている最小ローカルメモリサイズ以内に納める必要がある。プレーヤの最小ローカルメモリサイズが規定されていれば、アプリケーション制作者はプログラムのバイトコードのサイズ及びアプリケーションで利用

するイメージデータなどの合計サイズが、最小ローカルメモリサイズに収まるように調整することができ、サイズが決まれば作成後にベリファイすることも可能である。

【０１２９】

データ＃３はローカルメモリに余裕があれば読み込まれるデータ（４７０３）であり、これらのデータとしては、おまけのゲームや直接本編の再生やシナリオの分岐に関わらないアプリケーションであり、実行されなくてもメインコンテンツの再生に影響がないアプリケーションなどである。読み込みオプションのデータは、ローカルメモリに読み込み必須のデータを読み込んだ後にまだ空き容量がある場合、空き容量とデータサイズを比較し空き容量の方が大きければ、ディスクからローカルメモリに読み込まれる。メモリが少ない場合は読み込まれない。　図４８はデータ読み込みの優先度情報をさらに細かくしたものを示している。複数の読み込みオプションのデータがある場合、複数のデータを全て読み込めるだけのローカルメモリが搭載されていれば問題はない。複数存在するデータのうち一部のデータを読み込むだけのメモリ空き容量がなければ、どの読み込みオプションデータを読み込むか判断する必要がある。この順序は、プレーヤごとに違う動作をしてもよいが、データ管理テーブルにデータが出現する順序で読み込んでもよい。後者の場合、同じローカルメモリが搭載されていれば、同じデータが読み込まれることになる。

【０１３０】

あるいは、読み込みオプションのデータに対してはより細かい優先度（４８０１）を設け、メモリに空き容量がある場合、優先度が高いものからメモリに読み込めるか判定を行う。より優先度が高いデータを読み込める場合はそのデータを読み込み、さらに空き容量がある場合に限り次の優先度を持つデータを読み込めるか判定を行う。

図４９はオプションデータの読み込み判定のフローを示している。タイトルが遷移してデータ管理情報による読み込みが始まる（Ｓ４９０１）と、まず読み込み必須のデータをローカルメモリに読み込む（Ｓ４９０２）。読み込み必須データ以外に、読み込み判定を行っていない読み込みオプションデータがあるかチェックを行い（Ｓ４９０３）、読み込み判定を行っていないデータがあれば、判定を行っていないデータのうちもっとも優先度が高いデータに対して、ローカルメモリの空き容量とデータのサイズの比較を行う（Ｓ４９０４）。比較の結果、データサイズの方が大きければローカルメモリに読み込めないため、次の優先度のデータの判定に移行する。空き容量の方が大きければ、データをローカルメモリに読み込み（Ｓ４９０５）、次の優先度のデータの判定に移行する。全ての読み込みオプションデータの読み込みチェックが終了すれば、判定を終了する（Ｓ４９０６）。

（アプリケーション管理情報とデータ管理情報の統合）

アプリケーション管理情報はアプリケーションの有効期間を管理しており、ワークメモリ上のデータの生存期間を管理している。データ管理情報はデータの読み込み及び解放のタイミングを管理しており、ローカルメモリ上でのデータの生存期間を管理している。どちらも独立して管理可能であるが、アプリケーション及びデータの管理単位はタイトルなど同じ単位を基本としているため、生存期間が同じものは情報を結合させ、一元的に管理することも可能である。また、アプリケーションを実行するためにワークメモリにデータを読み込むためには、ローカルメモリにデータがなければならないので、アプリケーションに必要なデータは同じ生存期間で管理した方が簡潔である。

【０１３１】

図５０はアプリケーション管理情報にデータ管理情報を統合した、アプリケーション・データ管理情報を示している。アプリケーションは実行するために必要なデータと関連付けられるため、タイトル及びアプリケーションＩＤと読み込みデータを関連付けて管理情報を統合している。

なお管理情報が統合されたため、データの読み込み及び解放のタイミングや、ローカルメモリの制御、ディスクからローカルメモリへのデータの読み込み指示もアプリケーションマネージャが行ってもよいし、別のモジュールが管理してもよい。

【０１３２】

アプリケーション起動及び終了、データの読み込み・解放の基本的な管理単位はタイトルである。図中の例で説明する。タイトル＃１に切り替わりタイトルが開始されると、読み込み必須のデータをローカルメモリに読み込み、空き容量がある場合は読み込める分だけ読み込みオプションデータを読み込む。データを読み込み終わるとアプリケーションの起動属性にしたがい、アプリケーションマネージャがアプリケーションを起動させる。アプリケーション＃１はデータ読み込み優先度が読み込み必須であり、必ずローカルメモリにデータが読み込まれている。また、起動属性がＡｕｔｏＲｕｎであるため、タイトル開始時に自動的にアプリケーションが起動される。アプリケーション＃２は読み込みオプションであるため、メモリに余裕がある場合はローカルメモリに読み込まれているが、メモリが少ない場合はローカルメモリに読み込まれていない。

【０１３３】

図５１のアプリケーション・データ管理情報は、起動属性と読み込み優先度に相関関係を作り、より単純な静的データといくつかのルールにより読み込みの制御を実現する、別の形式の管理情報である。ルールの例としては、起動属性がＡｕｔｏＲｕｎのアプリケーションは、必ず指定の期間中起動されるとルールを決め、それらアプリケーションのデータは必ずローカルメモリに読み込まれることとする。また、ＡｕｔｏＲｕｎではないアプリケーションに関しては、有効期間が大きいものほど優先度が高い（図中の例ではアプリケーション＃３の方がアプリケーション＃４より優先度が高い）とし、優先順位にしたがってローカルメモリの空き容量に合わせて、該当するアプリケーションのデータを読み込む。

【０１３４】

なお、これらの優先順位は、起動属性に関係なく、アプリケーションの有効期間が大きいものほど、あるいは、アプリケーションの開始期間が早いものほど優先されるとしてもよいし、単に起動属性だけで優先順位付けしてもよい。また、アプリケーション・データ管理情報の登録順をデータの優先順位としてもよいし、それらの組み合わせでもよい。

特に映画コンテンツのような場合、ＡＶストリームを再生するためのアプリケーションは必ず実行されなければらないため、それら必ず起動されなければならないアプリケーションに対しては、起動属性としてさらに特別な属性（５１０１）を持たし、起動を保証するようにしてもよい。チャプターのようなＡＶストリームの時間軸上に定義されるような区間を利用する場合は、時間軸を決めるＡＶストリームを再生するアプリケーションが起動しなければ、ほかのアプリケーションを制御できなくなってしまうかもしれない。メモリの状態によりほかのアプリが起動されなくても、必ず起動されなければならないアプリケーションに、ＡｕｔｏＲｕｎ（ｍａｎｄａｔｏｒｙ）のような特別な起動属性を持たせると再生が保証される。

（アプリケーション起動時の例外処理）

全てのデータが必ず読み込まれているモデルではなく、ローカルメモリの容量に応じて読み込まれないデータが存在する場合、アプリケーションマネージャや他のアプリケーションからの制御により、ローカルメモリにデータがないアプリケーションの起動指示があり得る。

【０１３５】

図５２はデータの読み込み、アプリケーション起動の指示の様子を順を追って示している。アプリケーション・データ管理情報にしたがって、ローカルメモリにデータを読み込んでいく。このときデータ＃３（５２０１）は優先度が低く、ローカルメモリの容量が少なかったため読み込まれなかった。にもかかわらず、タイトル中でユーザーの操作などによりアプリケーション＃３の起動が促されると、Ｊａｖａ（登録商標）仮想マシンはワークメモリにデータを読み込むためにローカルメモリにアクセスする。このときのアプリケーションの読み込み命令に対し、ローカルメモリにデータがないことを確認したアプリケーションマネージャあるいはローカルメモリを管理するモジュールは、必要とされるデータの優先度を最大にし、使われておらず、読み込み必須ではないデータをローカルメモリ

中から削除してでも、必要なデータをディスクから読み出しにいくことも可能である。

アプリケーションがほかのアプリケーションの起動を指示した場合、ローカルディスクに起動しようとするアプリケーションのデータがなければ、アプリケーションの起動命令に対して次の３つのような動作が起こりえる。

1)アプリケーションは、ディスクからローカルメモリにデータが読み込まれ、ローカルメモリからワークメモリへのデータの読み込みが完了するまで待つ。ローカルメモリへのデータアクセスがアプリケーションから行われた結果、ディスクからローカルメモリへのデータの読み込みはアプリケーションの制御ではなく、アプリケーションマネージャあるいはローカルメモリを管理するモジュールが行う。

【０１３６】

2)起動命令（読み込み命令）にタイムアウトを設け、起動するまでに一定時間待機、その後それでもデータがローカルメモリから読み込めない場合は、読み込みに失敗と判断して、読み込み失敗時の例外処理を行う。

3)ローカルメモリにデータがない時点ですぐに読み込み失敗と判断、読み込み失敗時の例外処理を行う。

前述の３つのどの動作をするかは、アプリケーションのプログラム中にある、ほかのプログラム起動命令と合わせてプログラムしておけばよい。映画コンテンツの場合、ＡＶストリームの再生中に1)あるいは2)のタイプのデータの読み込みが発生すると、データを読み込むためにディスク上でシークが発生するかもしれない。その場合、ＡＶストリームの供給が一時的に止まるため、ＡＶストリームの再生が一時的に停止してしまうかもしれない。

【０１３７】

アプリケーションマネージャによるアプリケーション起動指示の場合、プログラムによる任意の例外処理はできないが、上記のどの方法をとることも可能である。ただし、アプリケーションマネージャが起動指示を出す性質上、アプリケーションのデータがローカルメモリに読み込まれていないのであれば、ローカルメモリの容量が足りなかった場合であり、ほかのデータを削除して起動するのはアプリケーション・データ管理情報の優先度情報と矛盾する動作である。そのため、2)あるいは3)の動作が望ましく、ＡＶストリームの再生中ならばＡＶストリームの再生を優先し、シークを発生させないことが望ましい。

【０１３８】

ローカルメモリが少ない場合は、アプリケーション起動のためデータ読み込みが発生しＡＶストリームの再生が一時的に停止する場合がある。ローカルメモリが多いとシークが発生せず、連続したＡＶストリームの再生が行える、あるいはレスポンスを早くできる利点がある。

ローカルメモリのサイズを固定せず拡張性を持たせる仕組みをアプリケーション・データ管理情報の優先度情報などの仕組みを提供することで、必要最低限のローカルメモリしか搭載していないプレーヤでも必ず再生あるいは実行されなければならないアプリケーションの動作保証を行い、大容量のローカルメモリを搭載したプレーヤはユーザーに快適なアプリケーションの動作を提供できる仕組みが実現可能となる。

（ＡＶストリームと同時にデータを提供する仕組み）

ここまで述べてきたデータ提供のタイミングは、タイトル開始時のＡＶストリームの再生が始まる前にデータをプリロードしておく方法、あるいは、アプリケーションがデータを実用としたときにＡＶストリームの再生を一時停止してでもデータを読みに行く方法であった。十分大容量のローカルメモリを搭載しておけば、タイトル選択中に必要となるデータを全てプリロードしておくことは可能であるが、必ずしも大容量を搭載しているわけではない。また、データを読み込むためにはディスク上でシークが発生してしまうため、

ＡＶストリームの再生が一時的に停止してしまう。そのため、図５３のような場合、アプリケーション・データ管理情報上は十分なローカルメモリを備えているにもかかわらず、有効に活用できなくなってしまう。

【０１３９】

データ＃１（５３０１）とデータ＃２（５３０２）は同時にローカルメモリに読み込むことが可能であり、データ＃１とデータ＃３（５３０３）も同時にローカルメモリに読み込むことが可能である。ただし、データ＃１とデータ＃２とデータ＃３を同時に読み込むだけの容量をローカルメモリは持っておらず、全てのデータをＡＶストリームの再生開始時（タイトル開始時）にプリロードすることはできない。そのため、チャプター＃２が終了してからチャプター＃４が開始されるまでの間に、データ＃２をローカルメモリから解放し、データ＃３をディスクから読み出してくる必要がある。ＡＶストリームと別ファイルで配置してあるとシークが発生してしまい、ＡＶストリームの再生が一時停止してしまうかもしれないが、ＡＶストリームが映画コンテンツなどの場合、再生が一時的に停止することは望ましくない。そこで、ＡＶストリームを再生中にシークをすることなく、データをローカルメモリに読み込む方法として、多重化方式あるいはインターリーブ方式を用いる。

【０１４０】

図５４は多重化方式で、データをＡＶストリームと一緒に多重化する方法である。多重化すべきＪａｖａ（登録商標）プログラムやイメージデータ（５４０１）は、データを管理する情報とともにセクション構造（５４０２）に内包する。セクションをＭＰＥＧストリーム多重化するために、セクション構造をＰＥＳパケット（５４０３）に内包する。ＰＥＳパケット（Ｐａｃｋｅｔｉｚｅｄ Ｅｌｅｍｅｎｔａｒｙ Ｓｔｒｅａｍパケット）は１８８バイト単位のＴＳパケット（Ｔｒａｎｓｐｏｒｔ Ｓｔｒｅａｍパケット、５４０４）に分割され、同様にＴＳパケットに分割されたビデオ（５４０５）及びオーディオ（５４０６）ストリームとともに多重化されて、１本のストリーム（５４０７）を構成する。

【０１４１】

このように多重化されたデータは、ストリームをＴＳパケットに分解し、ＰＥＳパケットに含まれるセクション、セクションに含まれるデータを抽出することで取り出すことが可能であり、また、ＡＶストリームと同時のデータを読み込むことが可能である。

図５５はデータをインターリーブ配置することにより、ＡＶデータと交互にデータを読み込む方法を示している。データ（５５０１）はデータの管理情報とともにインターリーブユニット（５５０２）に内包する。インターリーブユニットに内包するデータは１つでもよいし複数でもよい。インターリーブユニットはＡＶストリームファイル（５５０３）の途中に割り込ませて配置（インターリーブ配置）され、ＡＶストリームの開始点やＡＶストリームとインターリーブユニットとの境界点のアドレス情報を保持しておく。境界点情報から、どのアドレスのデータが、ＡＶストリームかデータかを判定することが可能である。データを読み込む際は、ＡＶストリームとインターリーブユニットを交互に続けて読み込み、読み込み中のアドレスがＡＶストリームの場合はＡＶストリームを処理する部分にデータを供給し、読み込み中のアドレスがインターリーブユニットを保持するアドレスであれば、インターリーブユニットを読み込み、内包されているデータを取り出して、ローカルメモリに格納する。

【０１４２】

このような方法により、ＡＶストリームの再生中であっても、データをローカルメモリに読み込むことが可能となり、図５３の例であってもチャプター＃３再生中に、シークを発生させることなく連続してデータを読み込み続けるだけで、データ＃３をローカルメモリに読み込むことが可能となる。

インターリーブ方式ではＡＶストリームとデータ（インターリーブユニット）の境界がはっきりしているため、シークによりデータだけ読み出すことは容易であるが、多重化方式では、ストリーム中にデータが分散して配置されてしまっているため、データだけ読み

込むためであったとしても、ストリーム全部を解析しデータ部分を取り出す必要があり、データの読み込みが非常に複雑な作業となる。そこで、ＡＶストリーム再生中はストリームとともに送られてくるデータを利用し、ローカルメモリに余裕がある場合にはＡＶストリームとは別のファイルとして配置されている同じデータからデータを読み出す仕組みを導入する。

【０１４３】

図５６は、ＡＶストリーム再生中にシークを発生させることなくデータを読み込むための多重化方式あるいはインターリーブ方式のデータと、ローカルメモリが大容量なためプリロードするためにＡＶストリームとは別ファイルで配置してある同じデータを管理するための、アプリケーション・データ管理情報の例を示している。ストリーム中に多重化あるいはインターリーブ配置されているデータの管理情報（５６０１）と別ファイルでは位置してあるデータの管理情報（５６０２）が２つ登録されているが、読み込まれるデータは同一である。

【０１４４】

ストリーム中のデータは、タイトル開始時あるいはＡＶストリーム開始時にディスクからローカルメモリにデータをプリロードすることはできない。優先度が読み込み必須であったとしても、プリロードするのではなく、データが必要になる前に、ＡＶストリーム再生していると自動的にローカルメモリに読み込まれる。この例ではアプリケーション＃３で必要になるデータはチャプター＃４以前に読み込まれていなければならないので、ストリーム中に埋め込まれたデータはチャプター＃４以前に配置しなければならない。データの配置は、一度だけではなく、複数回同じデータを埋め込むことにより、ＡＶストリームの再生を一部スキップしたとしても、再び出現したデータを読み込むことにより、データを読み込み損ねることを回避することも可能である。

【０１４５】

ローカルメモリが大容量な場合は、低い優先度のデータまでプリロードすることができるため、アプリケーション＃１及びアプリケーション＃２及びアプリケーション＃３のデータをすべてローカルメモリに読み込むことが可能となる。その場合は最初からアプリケーション＃３で必要なデータも読み込まれているため、ストリーム抽出する必要はなく、また、ＡＶストリームのジャンプにより、直接チャプター＃４に移動してもアプリケーションを起動することが可能となる。

（サイズが違う同じデータ名のデータの複数配置）

図５７は同じアプリケーションで必要なデータをローカルメモリのサイズに合わせて、イメージデータの解像度や品質を向上させる仕組みを、アプリケーション・データ管理情報で実現する方法を示している。

【０１４６】

アプリケーション＃１はプログラムとともにイメージデータを必要とするアプリケーションである。タイトル開始時にアプリケーション＃１用の読み込み必須データ（５７０１）をまずローカルメモリに読み込む。読み込み必須データは使用するイメージデータの品質を落とし、サイズを小さくしてある。ローカルメモリにまだ空き容量がある場合は、同じアプリケーション＃１用のデータで、次の優先度を持つもの（５７０２）を読み込もうとする。こちらのデータはイメージデータとしてそれなりの品質を持っているため、先ほどのイメージデータよりサイズが大きくなっている。ただし、これらのデータはディスク上のデータ名は異なっているが、ローカルメモリ上に展開される際は、同じファイル名になるように工夫してある。そのため、データが読み込まれることにより、品質の悪いイメージデータは品質がそれなりのイメージデータで上書きされる。同様に、より優先度は低いけれどイメージデータの品質がよくデータサイズが大きなデータ（５７０３）をローカルメモリに読み込むことが可能であれば、さらにイメージの品質を上げることが可能となる。このような管理方法により、ローカルメモリの容量が大きければ大きいほど、品質の高いアプリケーションを実行できる仕組みを提供可能となる。

【０１４７】
なお、同じファイル名のデータを複数回配置する場合、後から読み込まれたデータを優先して以前のデータを上書きしてもよいし、先に読み込まれたデータを優先して一度データを読むと削除されるまで同じファイル名のデータは読み込まないようにしてもよい。

（排他関係の明示）
アプリケーションによっては、排他関係にあるアプリケーションが存在するかもしれない。図５８の例のように、アプリケーション＃２とアプリケーション＃３はともにアプリケーション＃１より起動されるが、両方が同時に起動されることはない。このようなアプリケーションとして、アプリケーション＃１がアプリケーション起動ランチャー、アプリケーション＃２がゲーム、アプリケーション＃３が学習用アプリケーションのようなものがあげられる。アプリケーション起動ランチャーからはどちらか１つしか起動できず、どちらかのアプリケーションが起動中はランチャーが操作できないため、他方を起動することができない。
【０１４８】
このような場合、ローカルメモリ（５８０１）内は、アプリケーション＃１のデータとアプリケーション＃２のデータ、あるいは、アプリケーション＃１のデータとアプリケーション＃３のデータが共存することはあっても、３つのアプリケーションが同時にローカルメモリに読み込まれることはない。これらの状態を管理するために、アプリケーション・データ管理情報に互いにはいた関係にあるアプリケーションを識別するための情報を加えることにより、ローカルメモリのベリファイが容易になる。

（管理情報の配置）
アプリケーション・データ管理情報はディスク上のルートディレクトリに配置しておいてもよいし、タイトルごとに管理情報を分け、タイトルごとのディレクトリに配置してもよい。ＡＶストリームごとに管理情報を分け、ＡＶストリームの先頭に配置、あるいは、ストリームのヘッダ情報として埋め込んでおいてもよい。
【０１４９】
なお、ＡＶストリームのヘッダ情報としてアプリケーション・データ管理情報を配置する場合、優先度の判定はストリームが読み込まれるまで分からないため、タイトル開始時に優先度判定を行うのではなく、ストリームを読み込んでヘッダ情報からアプリケーション・データ管理情報を検出するたびに行ってもよい。（ＡＶストリームの再生の保証）
タイトルが選択されたときに、どのようにアプリケーションが起動されるか述べてきたが、アプリケーションがマスターとなってＡＶを再生するモデルであった。そのため、ＡＶ再生のマスターとなるアプリケーションが正常に動作するために、どのように動作保証するかが最大の問題となった。
【０１５０】
図５９はＡＶストリームの再生だけは、Ｊａｖａ（登録商標）仮想マシン上のアプリケーションで管理しない例を示している。タイトルリスト（５９０１）よりタイトルが選択されると、アプリケーションマネージャ（５９０２）がアプリケーション・データ管理情報（５９０３）を参照し、アプリケーションの起動及び終了を行う。アプリケーションマネージャによりＪａｖａ（登録商標）仮想マシン（５９０４）上に起動されたアプリケーションが再生制御部（５９０５）に指示を出すことにより、ＡＶストリームを再生することも可能である。タイトルリストの各タイトルはそのタイトル中で再生されるＰｌａｙＬｉｓｔ（５９０６）の情報を持っており、アプリケーションが起動しなくてもＪａｖａ（登録商標）仮想マシン上のアプリケーションとは別の再生制御モジュール（５９０７）によって、ＰｌａｙＬｉｓｔから参照されるＡＶストリームの再生を開始することも可能である。
【０１５１】
このような仕組みにより、たとえアプリケーションが起動しなかったりバグなどにより

暴走したとしても、ＡＶストリームの再生を担うモジュールは独立して動作し、ＡＶストリームの再生を保証することが可能となる。

なお、この場合は、再生制御モジュールが再生開始したＡＶストリームに対しても、アプリケーションから制御できることが望ましい。あるいは、アプリケーションが再生を開始したＡＶストリームであっても、何らかの異常でアプリケーションが停止してしまった場合、自動的に再生制御モジュールがＡＶ再生の制御を継承し、ＡＶストリームの再生を継続することも可能である。

【産業上の利用可能性】

【０１５２】

本発明によれば、映像データとプログラムとを含むコンテンツの開発にプログラミング環境を導入するため、特に、映像コンテンツの制作に携わる映画産業・民生機器産業において利用される可能性をもつ。

【図面の簡単な説明】

【０１５３】

【図１】ＤＶＤの構成図
【図２】ハイライトの構成図
【図３】ＤＶＤでの多重化の例を示す図
【図４】ＢＤ－ＲＯＭのデータ階層図
【図５】ＢＤ－ＲＯＭ上の論理空間の構成図
【図６】ＢＤ－ＲＯＭプレーヤの概要ブロック図
【図７】ＢＤ－ＲＯＭプレーヤの構成ブロック図
【図８】ＢＤ－ＲＯＭのアプリケーション空間の説明図
【図９】ＭＰＥＧストリーム（ＶＯＢ）の構成図
【図１０】パックの構成図
【図１１】ＡＶストリームとプレーヤ構成の関係を説明する図
【図１２】トラックバッファへのＡＶデータ連続供給モデル図
【図１３】ＶＯＢ情報ファイル構成図
【図１４】タイムマップの説明図
【図１５】タイムマップを使ったアドレス情報取得方法説明図
【図１６】プレイリストファイルの構成図
【図１７】プレイリストに対応するプログラムファイルの構成図
【図１８】ＢＤディスク全体管理情報ファイルの構成図
【図１９】グローバルイベントハンドラを記録するファイルの構成図
【図２０】タイムイベントの例を説明する図
【図２１】ユーザイベントの例を説明する図
【図２２】グローバルイベントハンドラの例を説明する図
【図２３】仮想マシンの構成図
【図２４】プレーヤ変数テーブルの図
【図２５】イベントハンドラ（タイムイベント）の例を示す図
【図２６】イベントハンドラ（ユーザイベント）の例を示す図
【図２７】プレーヤの基本処理のフローチャート
【図２８】プレイリスト再生処理のフローチャート
【図２９】イベント処理のフローチャート
【図３０】字幕処理のフローチャート
【図３１】プログラム実行環境を鑑みたプレーヤの概要ブロック図
【図３２】アプリケーション管理単位の説明図
【図３３】アプリケーション管理情報の説明図
【図３４】起動状態情報を備えたアプリケーション管理情報の説明図
【図３５】起動状態情報に休止状態を備えたアプリケーション管理情報の説明図
【図３６】起動属性によるタイトル間状態遷移の説明図

【図３７】階層的なアプリケーション管理単位の説明図
【図３８】チャプター単位の管理単位を備えたアプリケーション管理情報の説明図
【図３９】データ及びアプリケーションの管理に関わるプレーヤの概要ブロック図
【図４０】アプリケーションの起動及び終了の説明図
【図４１】イベントを利用したアプリケーション終了方法の説明図
【図４２】データの読み込み及び解放タイミングの説明図
【図４３】ディスク上とメモリ上のデータを関連付けるデータ管理情報を示す図
【図４４】複数ファイルを結合して管理するデータ構造の説明図
【図４５】アプリケーション管理とデータの読み込みのタイミングの関連の説明図
【図４６】メモリサイズの拡張性と動作保証についての説明図
【図４７】データ読み込み優先度情報を備えたアプリケーション管理情報を示す図
【図４８】細かい区分の優先度情報を備えたアプリケーション管理情報を示す図
【図４９】優先度情報を用いたデータ読み込み判定のフローチャート
【図５０】アプリケーションとデータの管理を統合した管理情報を示す図
【図５１】データ読み込み優先度情報をアプリケーション起動属性情報に従属させた管理情報を示す図
【図５２】アプリケーション起動の流れの説明図
【図５３】プリロードのみではメモリを有効に使えない例を示す図
【図５４】データをＡＶストリームとともに多重化した構造図
【図５５】データをＡＶストリームにインタリーブ配置した構造図
【図５６】ストリーム中のデータと別ファイルのデータを扱った管理情報を示す図
【図５７】優先度情報利用しメモリサイズに応じてデータの品質を向上する説明図
【図５８】排他的に起動されるアプリケーションの識別情報を備えた管理情報を示す図
【図５９】プログラム実行部とＡＶストリーム再生部を分けた概要ブロック図

【符号の説明】

【０１５４】

２０１　ＢＤディスク
２０２　光ピックアップ
２０３　プログラム記録メモリ
２０４　管理情報記録メモリ
２０５　ＡＶ記録メモリ
２０６　プログラム処理部
２０７　管理情報処理部
２０８　プレゼンテーション処理部
２０９　イメージプレーン
２１０　ビデオプレーン
２１１　合成処理部

３０１　プログラム記録メモリ
３０２　プログラムプロセッサ
３０３　ＵＯＰマネージャ
３０４　管理情報記録メモリ
３０５　シナリオプロセッサ
３０６　プレゼンテーションコントローラ
３０７　クロック
３０８　イメージメモリ
３０９　トラックバッファ
３１０　デマルチプレクサ
３１１　イメージプロセッサ

３１２　ビデオプロセッサ
３１３　サウンドプロセッサ
３１４　イメージプレーン
３１５　ビデオプレーン
３１６　合成処理部
３１７　ドライブコントローラ

３１０１　プログラム
３１０２　プログラム処理部
３１０３　プログラム処理部から管理情報処理部の制御
３１０４　イメージプレーン
３１０５　プログラム処理部からイメージプレーンの制御

３２０１　タイトルリスト
３２０２　本編映像データを再生するアプリケーション
３２０３　ポップアップメニューを表示するアプリケーション
３２０４　オンラインショッピング購入品を蓄えておくアプリケーション
３２０５　オンラインショッピングを実現するアプリケーション
３２０６　ゲームアプリケーション

３３０１　アプリケーション管理情報
３３０２　タイトル番号
３３０３　アプリケーションＩＤ
３３０４　アプリケーション＃１の起動
３３０５　アプリケーション＃１の終了
３３０６　アプリケーション＃３の起動

３４０１　起動属性
３４０２　アプリケーション＃１によるアプリケーション＃２の起動
３４０３　タイトル終端におけるアプリケーションの終了
３４０４　アプリケーション＃２によるアプリケーション＃１の終了

３５０１　タイトル＃１におけるアプリケーション＃１の登録情報
３５０２　タイトル＃２におけるアプリケーション＃１の登録情報

３７０１　タイトル単位のアプリケーション管理情報
３７０２　ディスク単位のアプリケーション管理情報
３７０３　ディスクセット単位のアプリケーション管理情報

３８０１　チャプター
３８０２　チャプター単位で管理されたアプリケーション管理情報
３８０３　チャプター単位で管理されたアプリケーション

３９０１　ディスク
３９０２　ローカルメモリ
３９０３　Ｊａｖａ（登録商標）仮想マシン
３９０４　ワークメモリ
３９０５　Ｊａｖａ（登録商標）アプリケーション
３９０６　再生制御部
３９０７　アプリケーションマネージャ

４００１　アプリケーションマネージャによるタイトル開始時の起動
４００２　アプリケーションマネージャによるイベント発生時の起動
４００３　アプリケーションによる起動
４００４　アプリケーションマネージャによるタイトル終了時の終了
４００５　アプリケーションマネージャによるイベント発生時の終了
４００６　アプリケーションによる終了

４１０１　終了イベント受信部
４１０２　終了プロセス
４１０３　終了イベント及び終了プロセスを持たないアプリケーション

４５０１　データ管理情報
４５０２　タイトル番号
４５０３　読み込みデータ名
４５０４　ディスクからローカルメモリへのデータの読み込み
４５０５　ローカルメモリからデータの解放

４７０１　読み込み優先度情報
４７０２　必ずローカルメモリに読み込まれるデータ
４７０３　必ずしもローカルメモリには読み込まれないデータ

４８０１　細かい区分の読み込み優先度情報

５１０１　起動属性に統合された読み込み必須属性

５２０１　読み込み優先度の低いデータ

５３０１　タイトル全体で有効なデータ
５３０２　チャプター＃１から＃２の間で有効なデータ
５３０３　チャプター＃４から＃５の間で有効なデータ

５４０１　データ
５４０２　セクション
５４０３　Ｐａｃｋｅｔｉｚｅｄ　Ｅｌｅｍｅｎｔａｒｙ　Ｓｔｒｅａｍ
　　パケット
５４０４　Ｔｒａｎｓｐｏｒｔ　Ｓｔｒｅａｍ パケット
５４０５　ビデオデータ
５４０６　オーディオデータ
５４０７　多重化されたストリーム

５５０１　データ
５５０２　インターリーブユニット
５５０３　ＡＶストリーム
５６０１　ストリーム中に多重化あるいはインターリーブ配置
　　されたデータ
５６０２　ストリームとは別ファイルとして配置されたデータ

５７０１　優先度が高くサイズが小さいデータ
５７０２　優先度が並でサイズが並のデータ

５７０３　優先度が低くサイズが大きいデータ

５８０１　ローカルメモリ

５９０１　タイトルリスト
５９０２　アプリケーションマネージャ
５９０３　アプリケーション・データ管理情報
５９０４　Ｊａｖａ（登録商標）仮想マシン
５９０５　再生制御部
５９０６　プレイリスト
５９０７　再生制御モジュール

Ｓ１０１　ディスク挿入ステップ
Ｓ１０２　ＢＤ．ＩＮＦＯ読み込みステップ
Ｓ１０３　ＢＤ．ＰＲＯＧ読み込みステップ
Ｓ１０４　ファーストイベント生成ステップ
Ｓ１０５　イベントハンドラ実行ステップ

Ｓ２０１　ＵＯＰ受付ステップ
Ｓ２０２　ＵＯＰイベント生成ステップ
Ｓ２０３　メニューコール判定ステップ
Ｓ２０４　イベント生成ステップ
Ｓ２０５　イベントハンドラ実行ステップ

Ｓ３０１　プレイリスト再生開始ステップ
Ｓ３０２　プレイリスト情報（ＸＸＸ．ＰＬ）読み込みステップ
Ｓ３０３　プレイリストプログラム（ＸＸＸ．ＰＲＯＧ）
　　読み込みステップ
Ｓ３０４　セル再生開始ステップ
Ｓ３０５　ＡＶ再生開始ステップ

Ｓ４０１　ＡＶ再生開始ステップ
Ｓ４０２　ＶＯＢ情報（ＹＹＹ．ＶＯＢＩ）読み込みステップ
Ｓ４０３　ＶＯＢ（ＹＹＹ．ＶＯＢ）読み込みステップ
Ｓ４０４　ＶＯＢ再生開始ステップ
Ｓ４０５　ＶＯＢ再生終了ステップ
Ｓ４０６　次セル存在判定ステップ

Ｓ５０１　プレイリスト再生開始ステップ
Ｓ５０２　プレイリスト再生終了判定ステップ
Ｓ５０３　タイムイベント時刻判定ステップ
Ｓ５０４　イベント生成ステップ
Ｓ５０５　イベントハンドラ実行ステップ

Ｓ６０１　プレイリスト再生開始ステップ
Ｓ６０２　プレイリスト再生終了判定ステップ
Ｓ６０３　ＵＯＰ受付判定ステップ
Ｓ６０４　ＵＯＰイベント生成ステップ
Ｓ６０５　メニューコール判定ステップ
Ｓ６０６　ユーザーイベント有効期間判定ステップ

Ｓ６０７　イベント生成ステップ
Ｓ６０８　イベントハンドラ実行ステップ

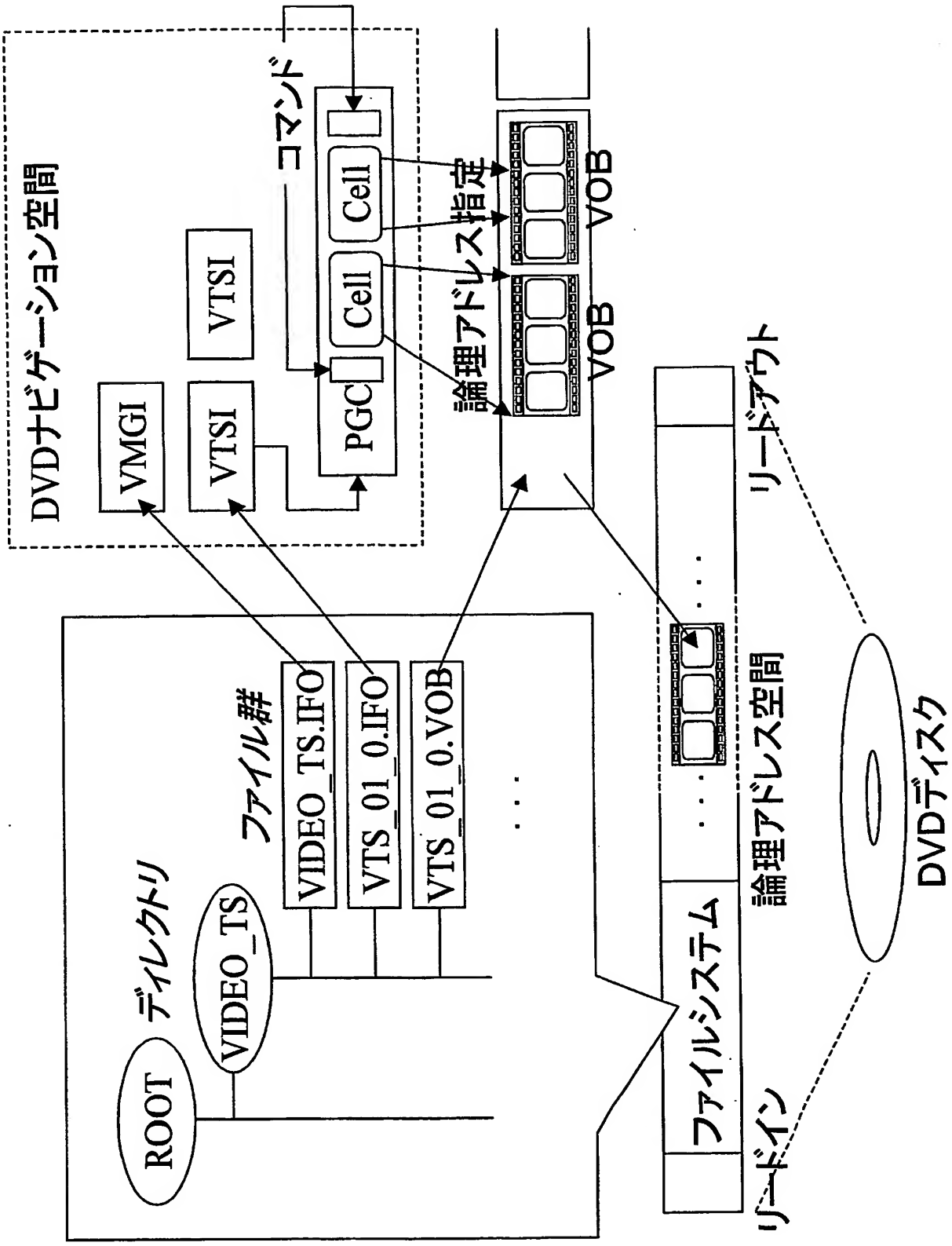
Ｓ７０１　プレイリスト再生開始ステップ
Ｓ７０２　プレイリスト再生終了判定ステップ
Ｓ７０３　字幕描画開始判定ステップ
Ｓ７０４　字幕描画ステップ
Ｓ７０５　字幕表示終了判定ステップ
Ｓ７０６　字幕消去ステップ

Ｓ１００１　ディスク挿入ステップ
Ｓ１００２　ＢＤ．ＣＬＡＳＳ検出判定ステップ
Ｓ１００３　ＢＤオブジェクト生成ステップ
Ｓ１００４　イベントハンドラ宣言ステップ
Ｓ１００５　ＢＤ．ＩＮＦＯ読み取りステップ
Ｓ１００６　ファーストイベント生成ステップ
Ｓ１００７　イベントハンドラ処理ステップ

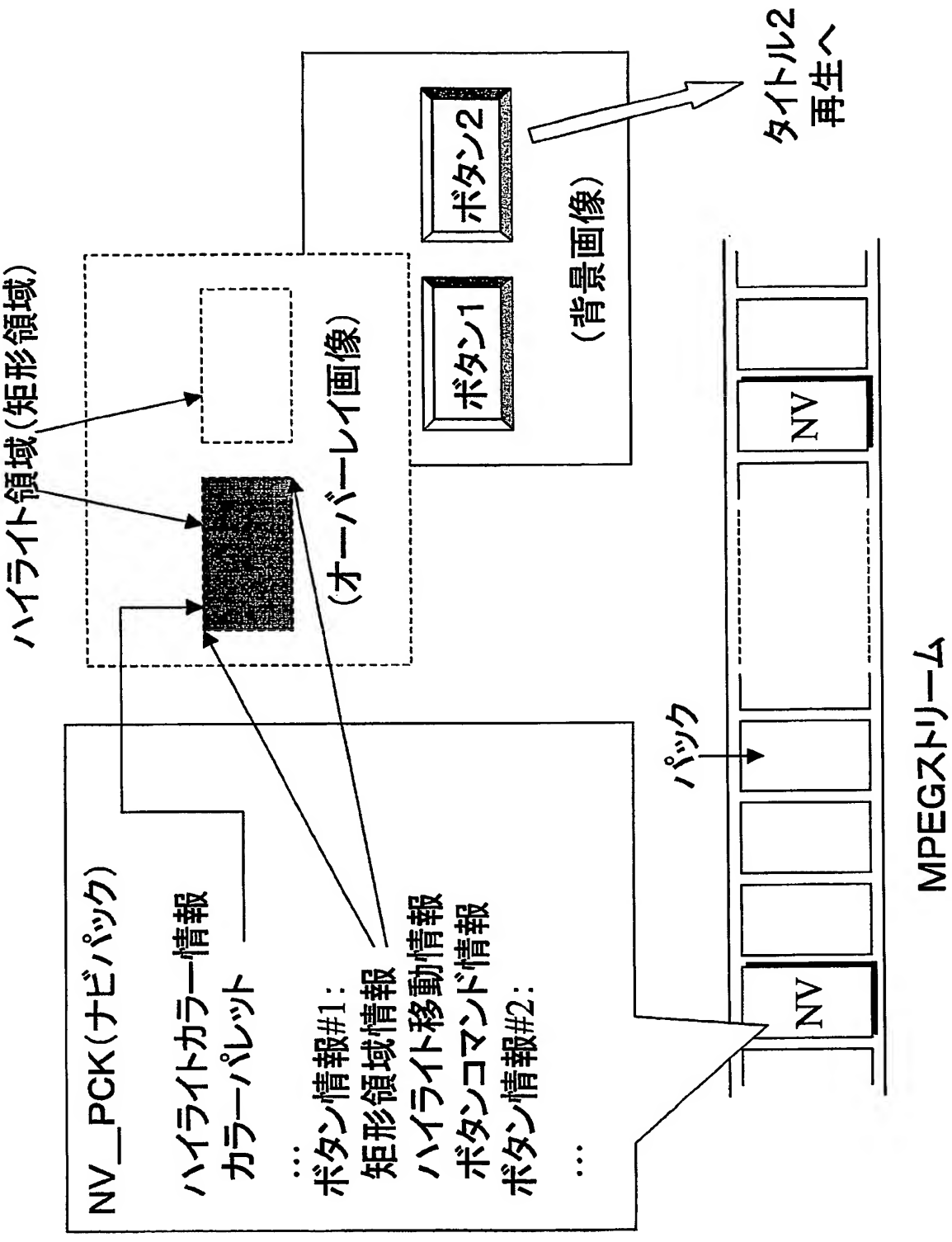
Ｓ１１０１　プレイリスト再生開始ステップ
Ｓ１１０２　プレイリスト終了判定ステップ
Ｓ１１０３　タイムイベント時刻判定ステップ
Ｓ１１０４　タイムイベント生成ステップ
Ｓ１１０５　イベントハンドラ処理ステップ

Ｓ１２０１　プレイリスト再生開始ステップ
Ｓ１２０２　モード切替判定ステップ
Ｓ１２０３　現有効モード判定ステップ
Ｓ１２０４　Ｊａｖａ（登録商標）モードへの切り替えステップ
Ｓ１２０５　ＤＶＤ互換モードへの切り替えステップ
Ｓ４９０１　判定開始ステップ
Ｓ４９０２　読み込み必須データの読み込みステップ
Ｓ４９０３　読み込みオプションデータの優先度判定ステップ
Ｓ４９０４　ローカルメモリの空き容量とデータサイズの比較ステップ
Ｓ４９０５　オプションデータの読み込みステップ
Ｓ４９０６　判定終了ステップ

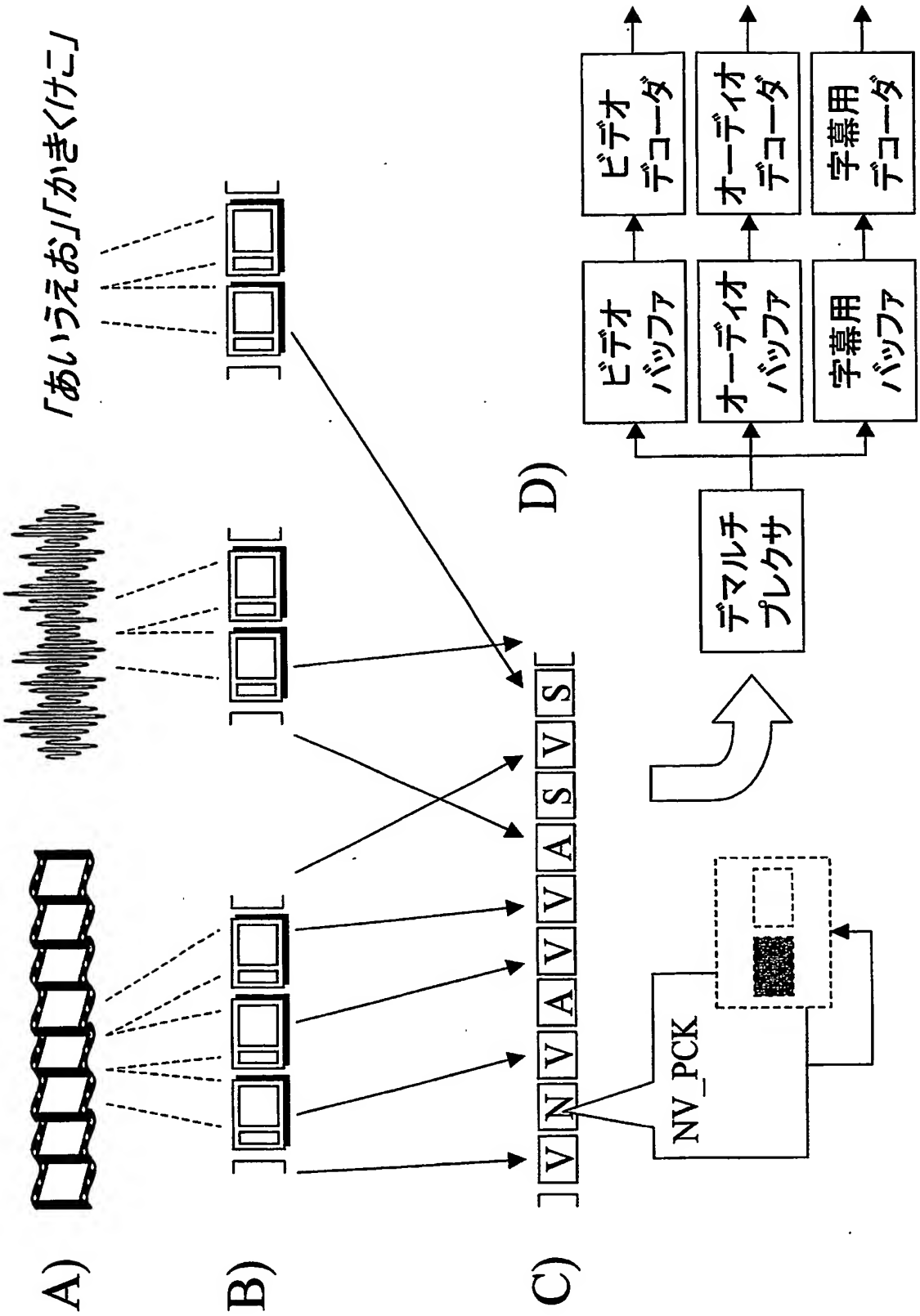
【書類名】図面
【図１】


DVDナビゲーション空間
VMGI
VTSI
VTSI
PGC
コマンド
Cell
Cell
論理アドレス指定
VOB
VOB
リードアウト
ROOT
ディレクトリ
VIDEO_TS
ファイル群
VIDEO_TS.IFO
VTS_01_0.IFO
VTS_01_0.VOB
ファイルシステム
論理アドレス空間
DVDディスク
リードイン

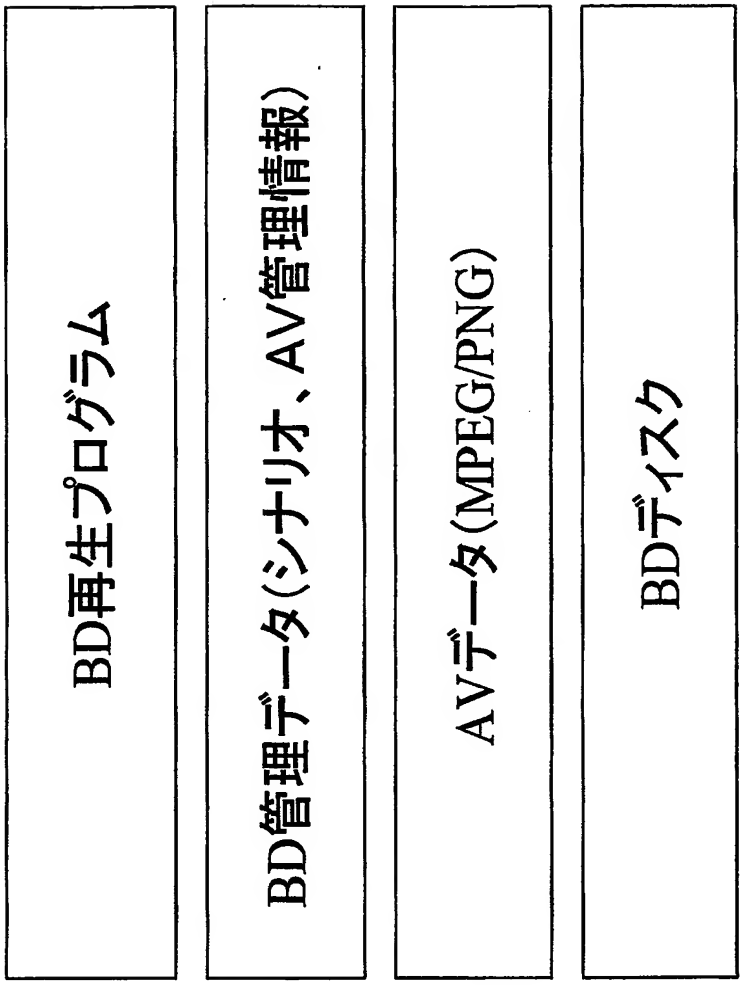
【図２】


ハイライト領域(矩形領域)
(オーバーレイ画像)
ボタン1
ボタン2
(背景画像)
タイトル2
再生へ
NV_PCK(ナビパック)
ハイライトカラー情報
カラーパレット
…
ボタン情報#1:
矩形領域情報
ハイライト移動情報
ボタンコマンド情報
ボタン情報#2:
…
パック
NV
NV
MPEGストリーム

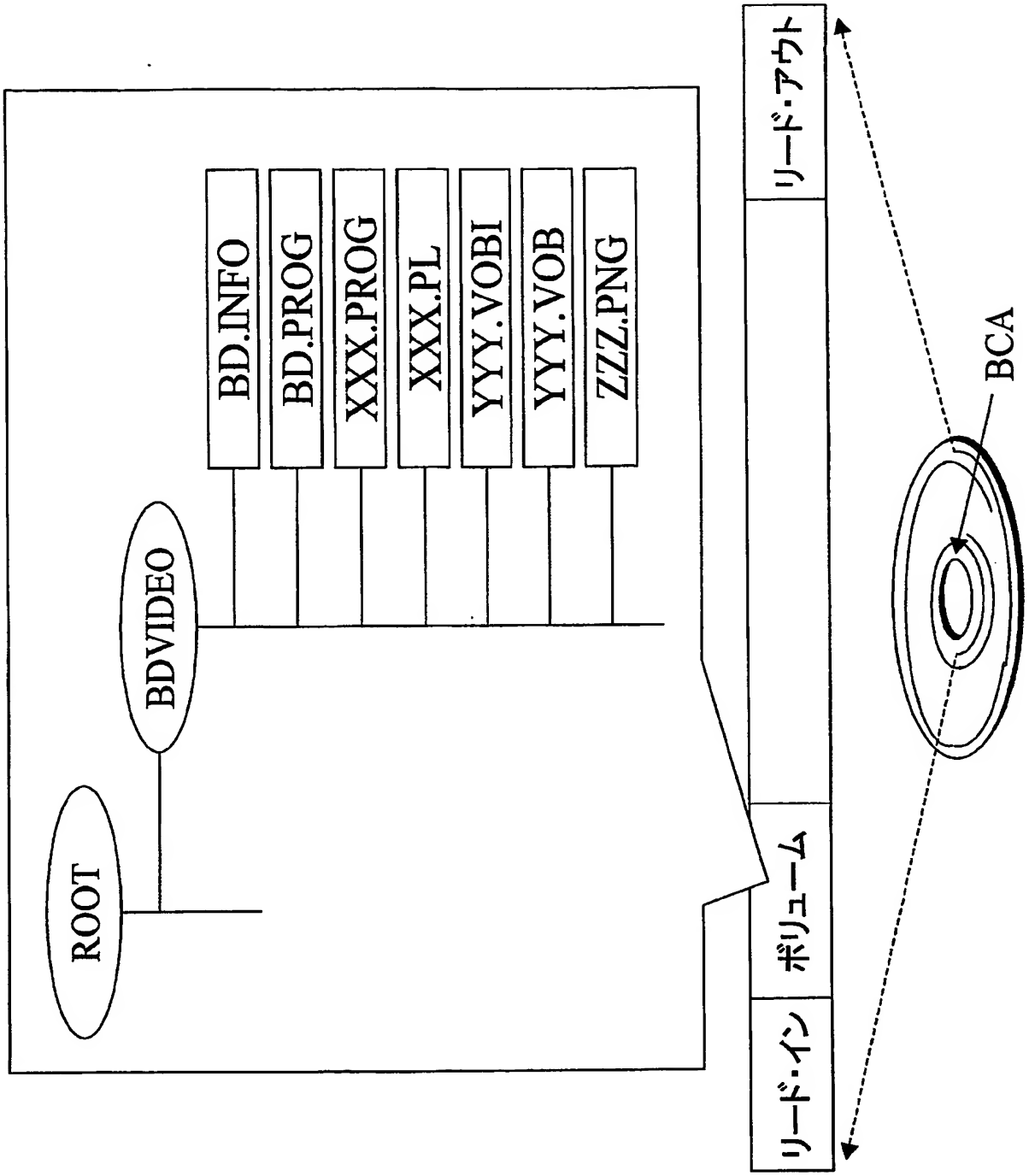
【図３】


「あいうえお」「かきくけこ」
A)
B)
C)
D)
V
N
V
A
V
V
A
S
V
S
NV_PCK
デマルチプレクサ
ビデオバッファ
オーディオバッファ
字幕用バッファ
ビデオデコーダ
オーディオデコーダ
字幕用デコーダ

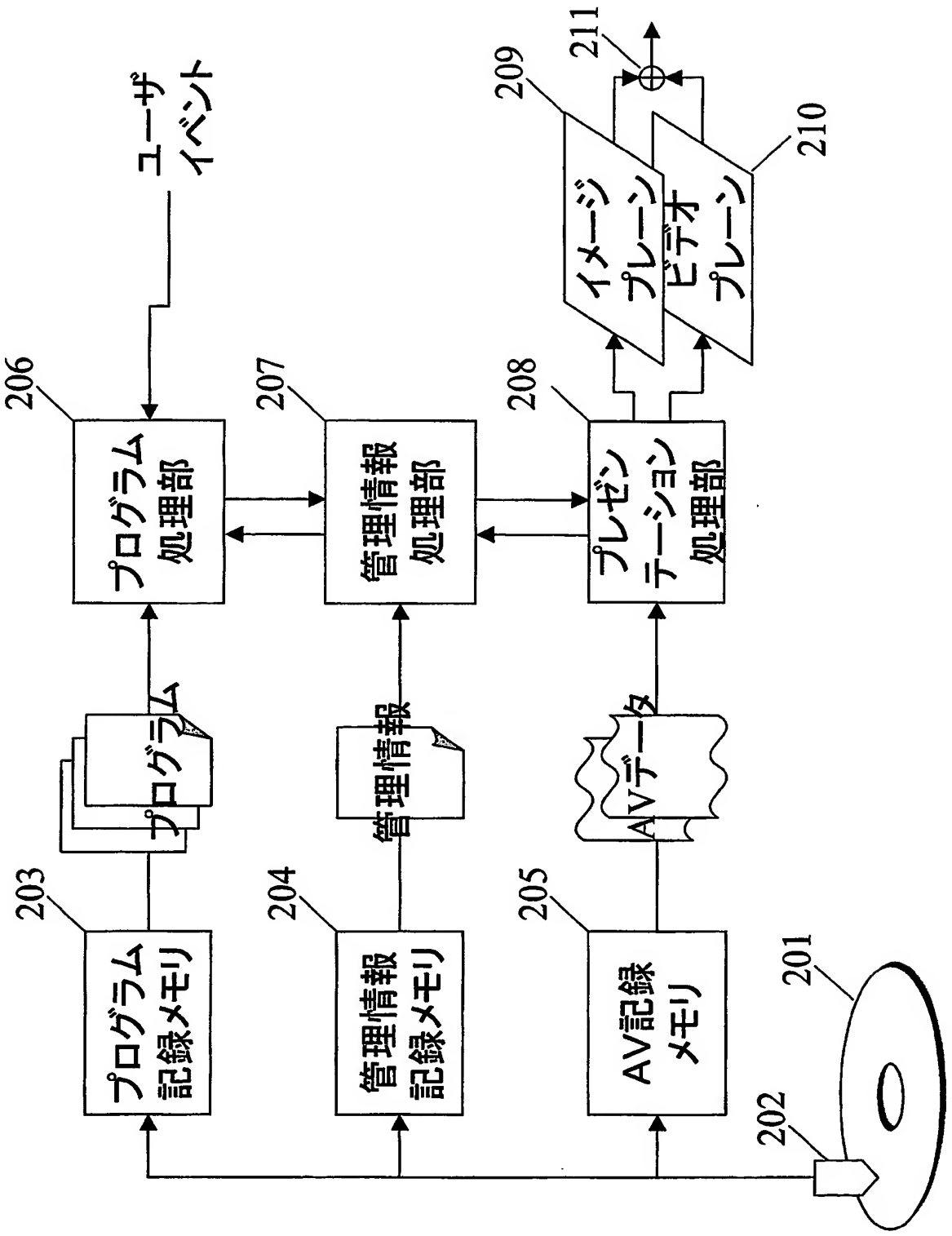
【図４】



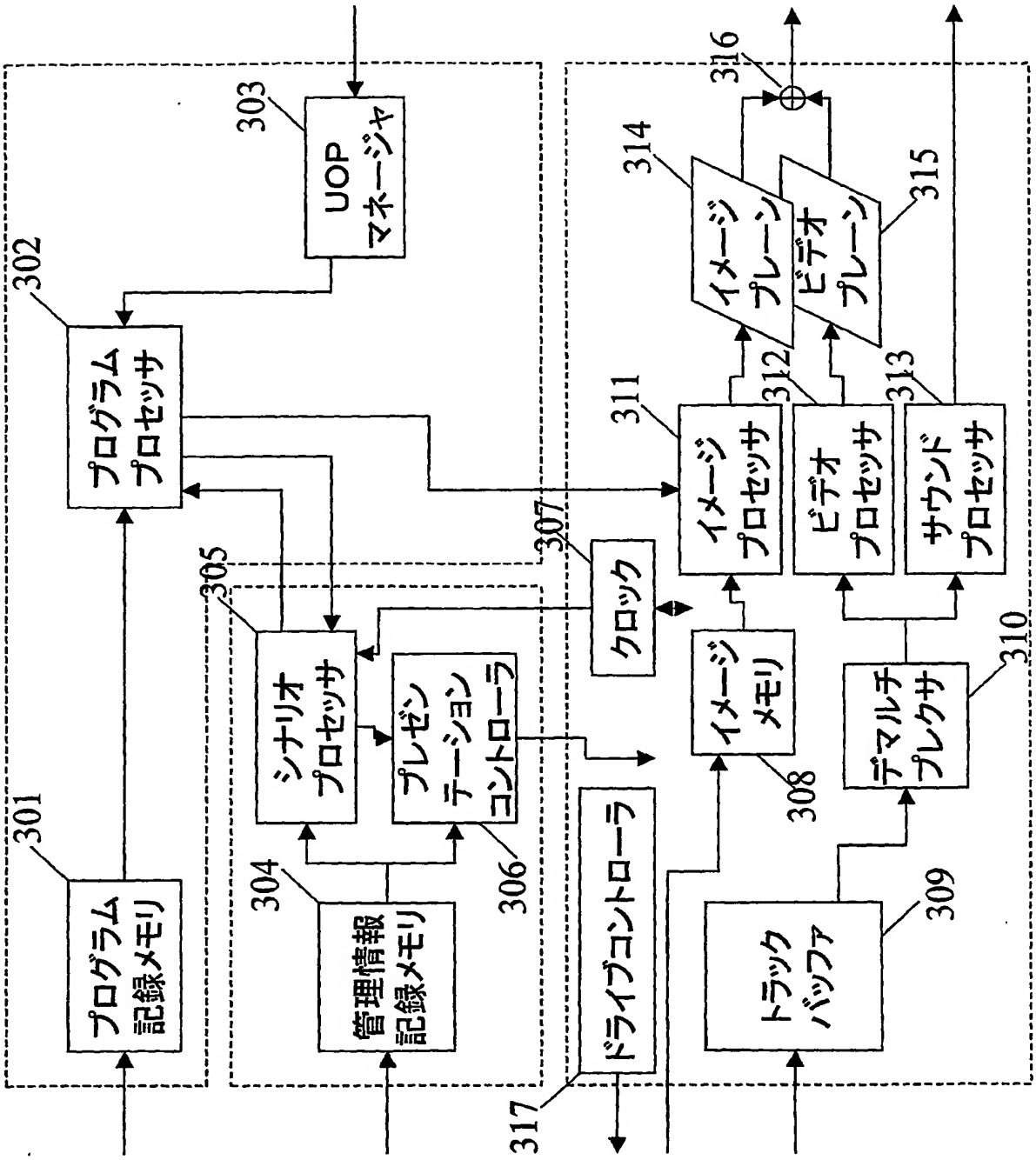
【図５】


ROOT
BDVIDEO
BD.INFO
BD.PROG
XXX.PROG
XXX.PL
YYY.VOBI
YYY.VOB
ZZZ.PNG
リード・イン
ボリューム
リード・アウト
BCA

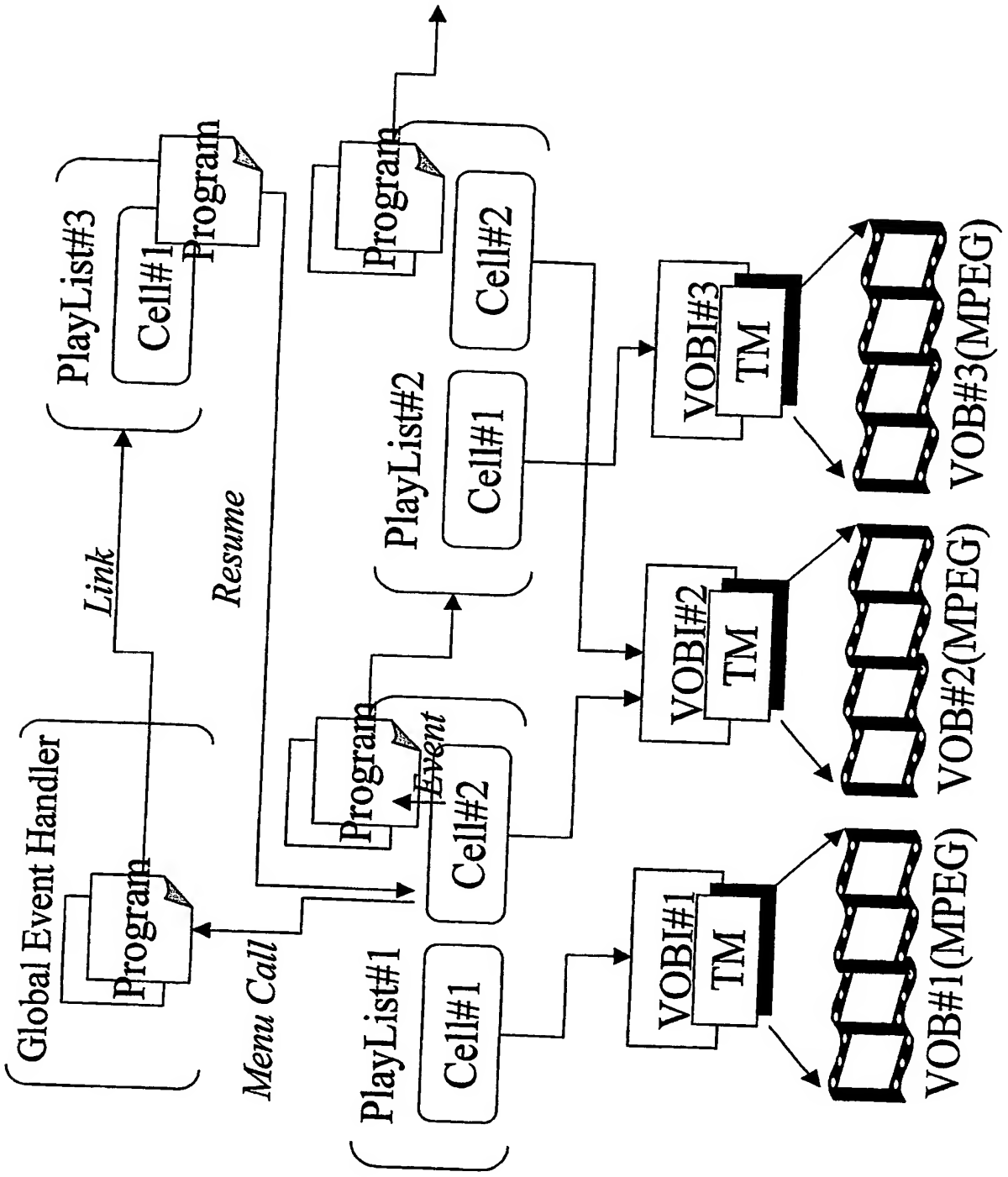
【図６】


ユーザ
イベント
206
プログラム
処理部
207
管理情報
処理部
208
プレゼン
テーション
処理部
209
211
210
イメージ
プレーン
ビデオ
プレーン
プログラム
管理情報
AVデータ
203
プログラム
記録メモリ
204
管理情報
記録メモリ
205
AV記録
メモリ
201
202

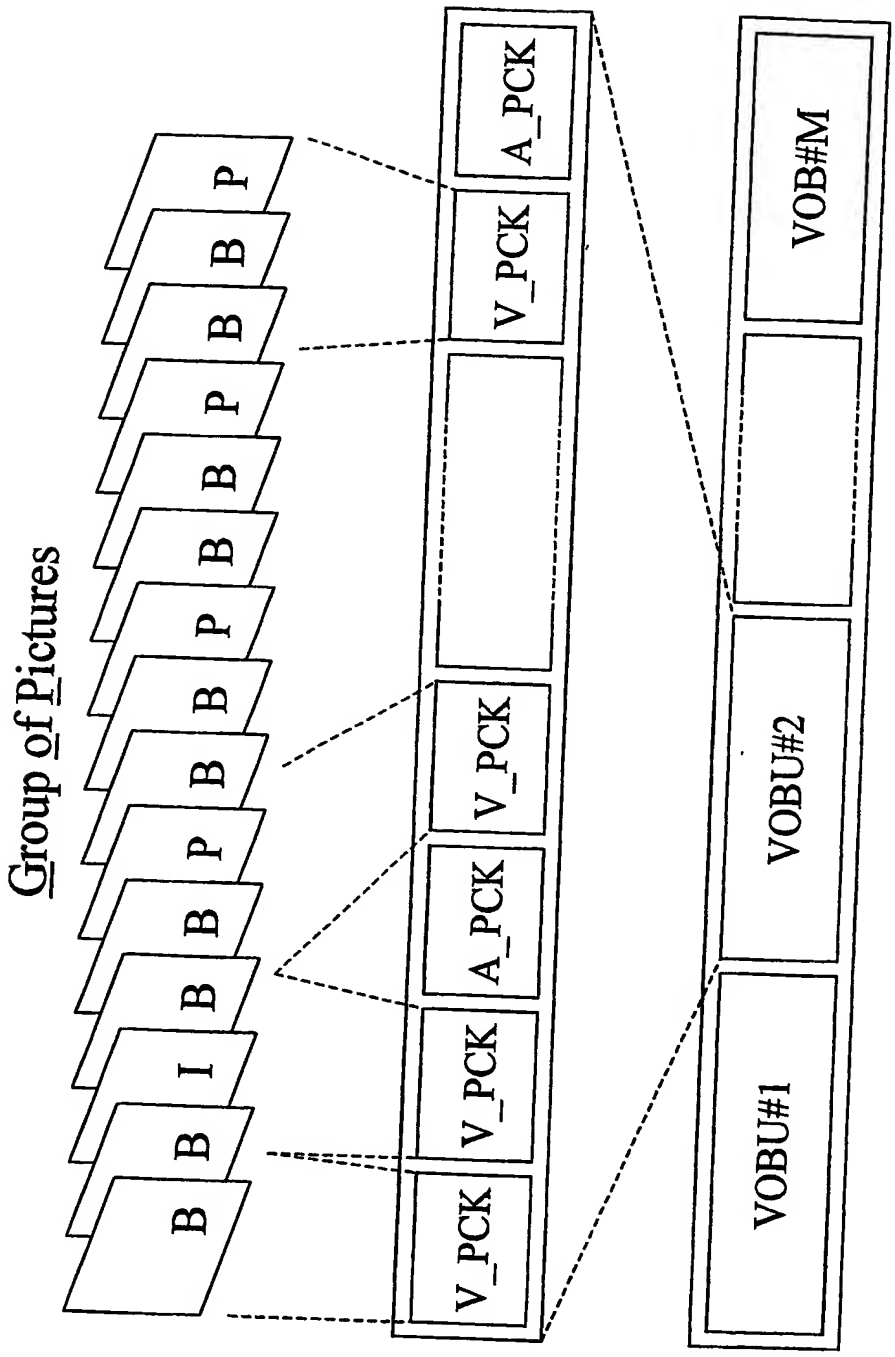
【図７】


301
プログラム記録メモリ
302
プログラムプロセッサ
303
UOPマネージャ
304
管理情報記録メモリ
305
シナリオプロセッサ
306
プレゼンテーションコントローラ
307
クロック
308
イメージメモリ
309
トラックバッファ
310
デマルチプレクサ
311
イメージプロセッサ
312
ビデオプロセッサ
313
サウンドプロセッサ
314
イメージプレーン
315
ビデオプレーン
316
317
ドライブコントローラ

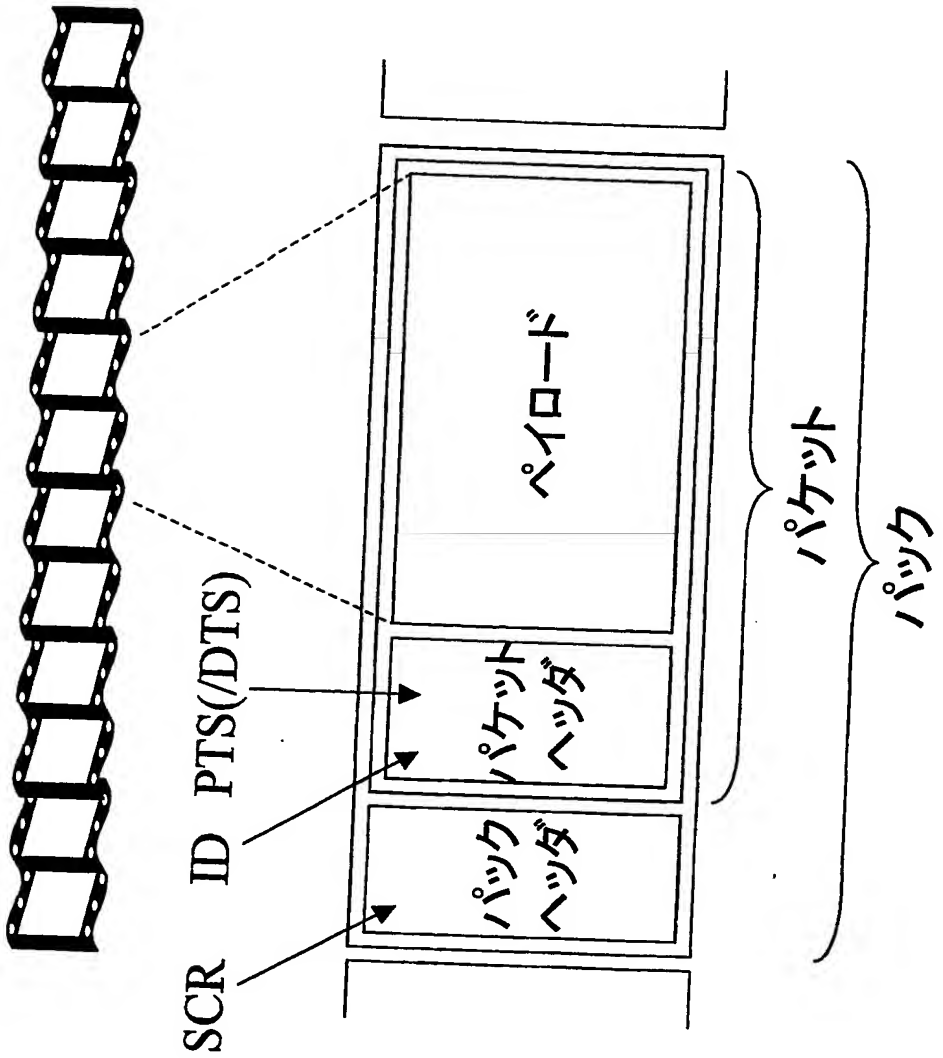
【図８】


Global Event Handler
Program
Menu Call
Link
Resume
PlayList#3
Cell#1
Program
PlayList#1
Cell#1
PlayList#2
Cell#1
Program
Event
Cell#2
Program
Cell#2
VOBI#1
TM
VOBI#2
TM
VOBI#3
TM
VOB#1(MPEG)
VOB#2(MPEG)
VOB#3(MPEG)

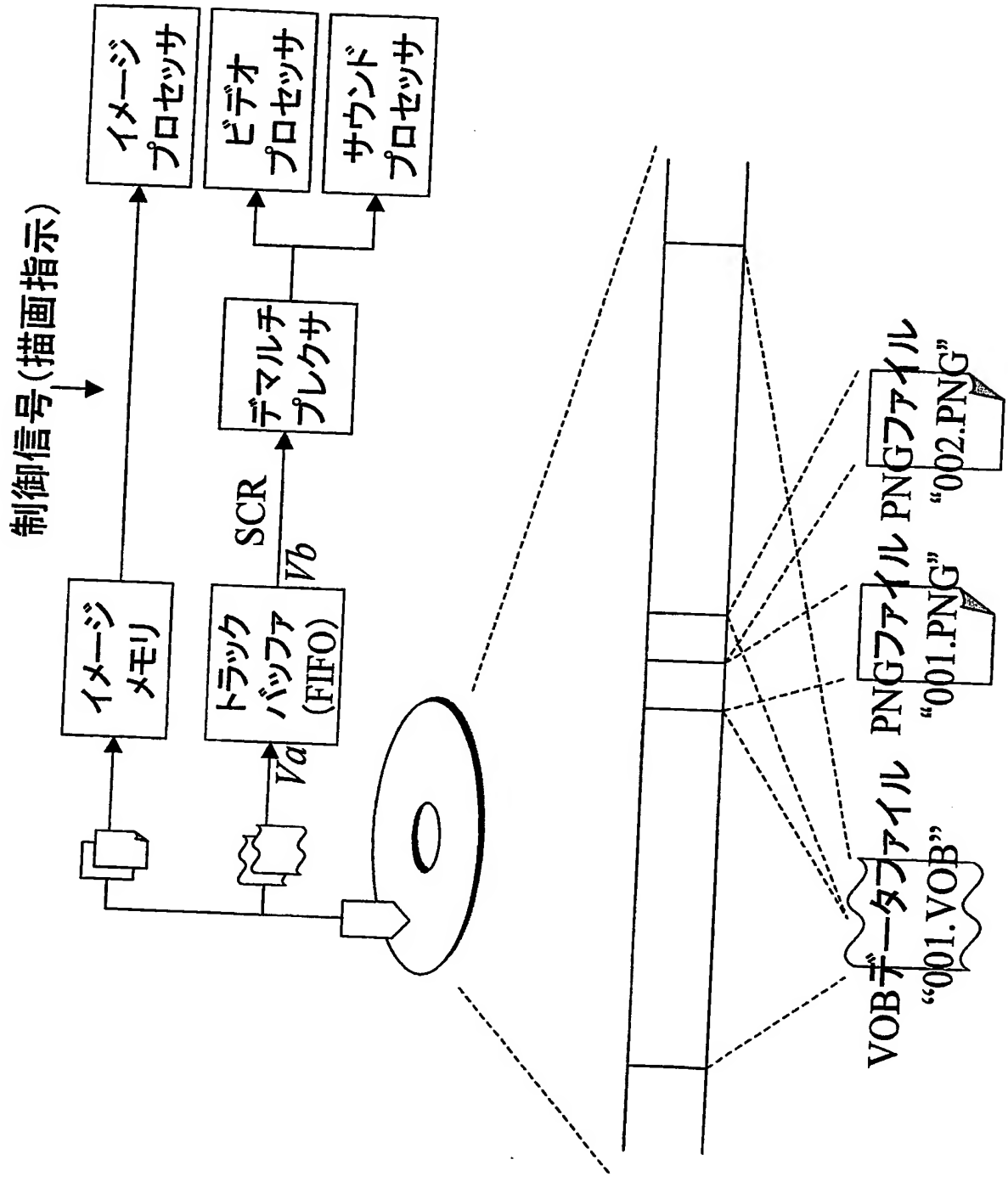
【図９】


Group of Pictures
B
B
I
B
P
B
B
P
B
B
P
B
B
P
B
A_PCK
V_PCK
V_PCK
A_PCK
V_PCK
V_PCK
VOBU#1
VOBU#2
VOB#M

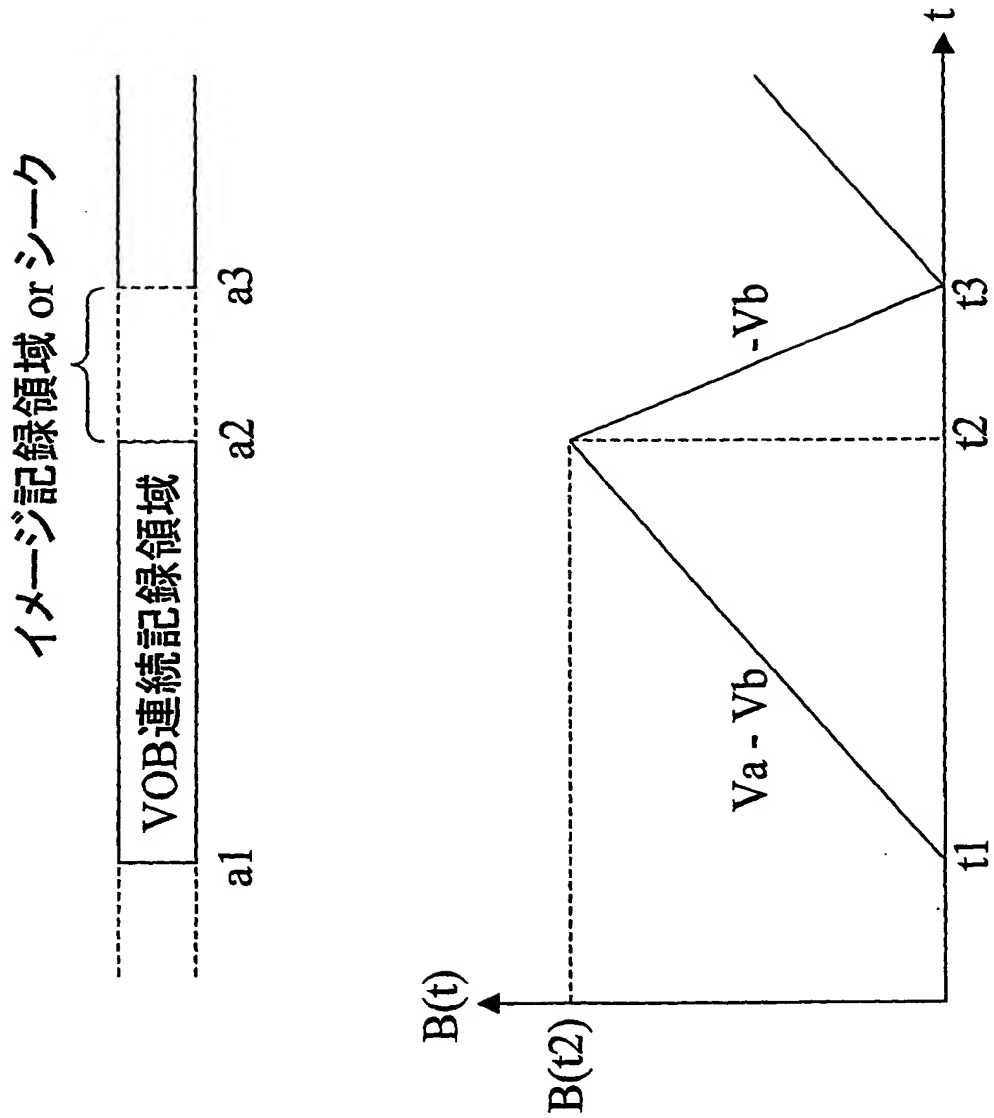
【図１０】



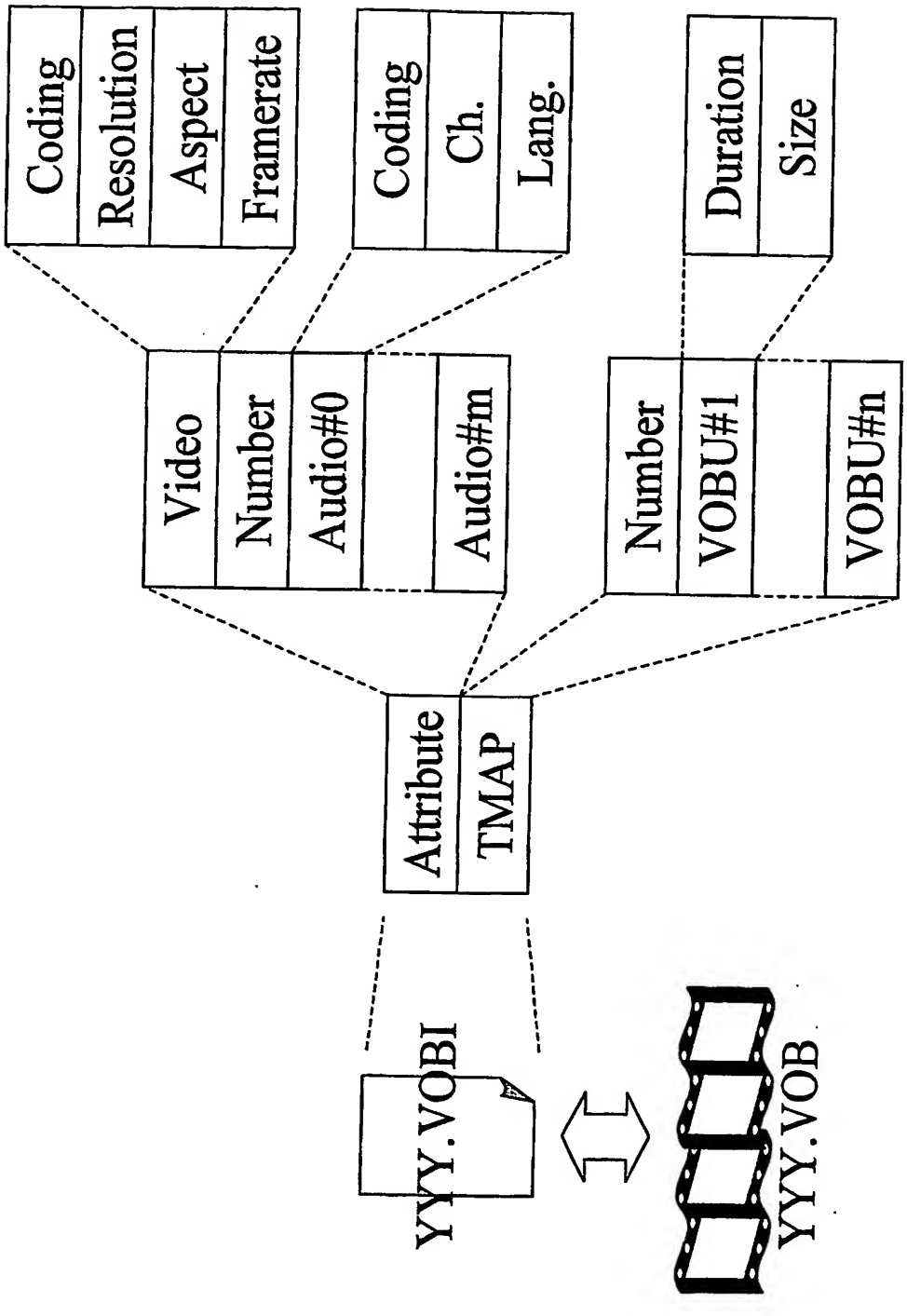
【図１１】


イメージプロセッサ
ビデオプロセッサ
サウンドプロセッサ
制御信号(描画指示)
デマルチプレクサ
SCR
Vb
Va
イメージメモリ
トラックバッファ(FIFO)
VOBデータファイル "001.VOB"
PNGファイル "001.PNG"
PNGファイル "002.PNG"

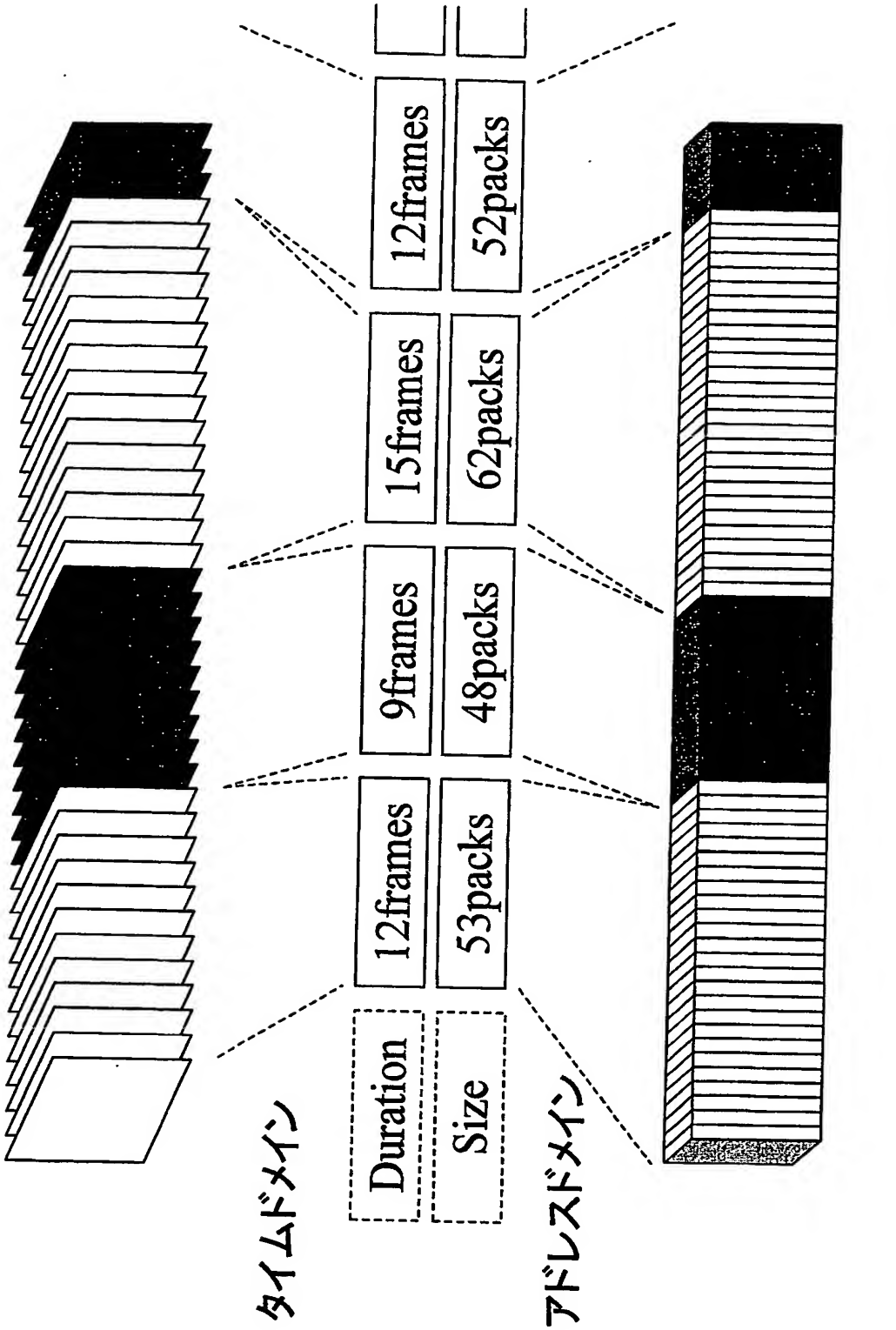
【図１２】



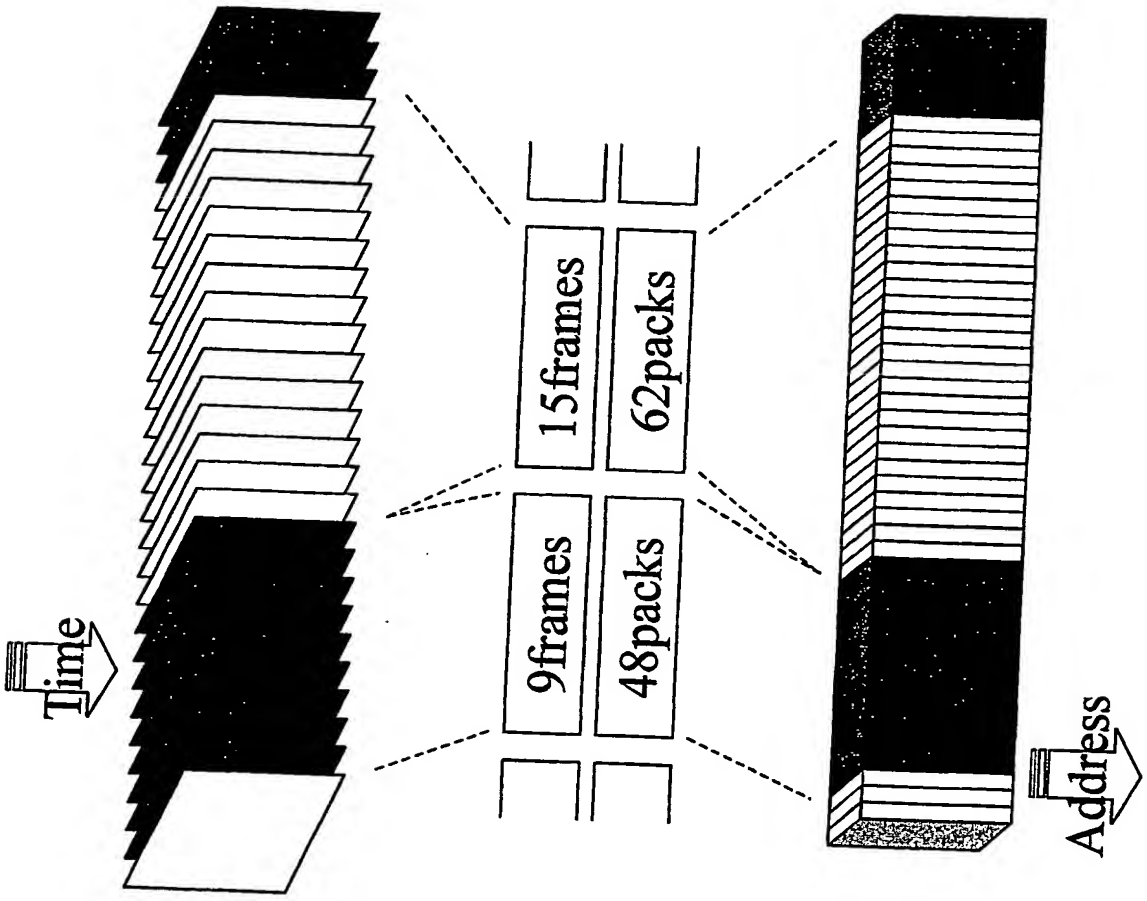
【図１３】


Coding
Resolution
Aspect
Framerate
Coding
Ch.
Lang.
Duration
Size
Video
Number
Audio#0
Audio#m
Number
VOBU#1
VOBU#n
Attribute
TMAP
YYY.VOBI
YYY.VOB

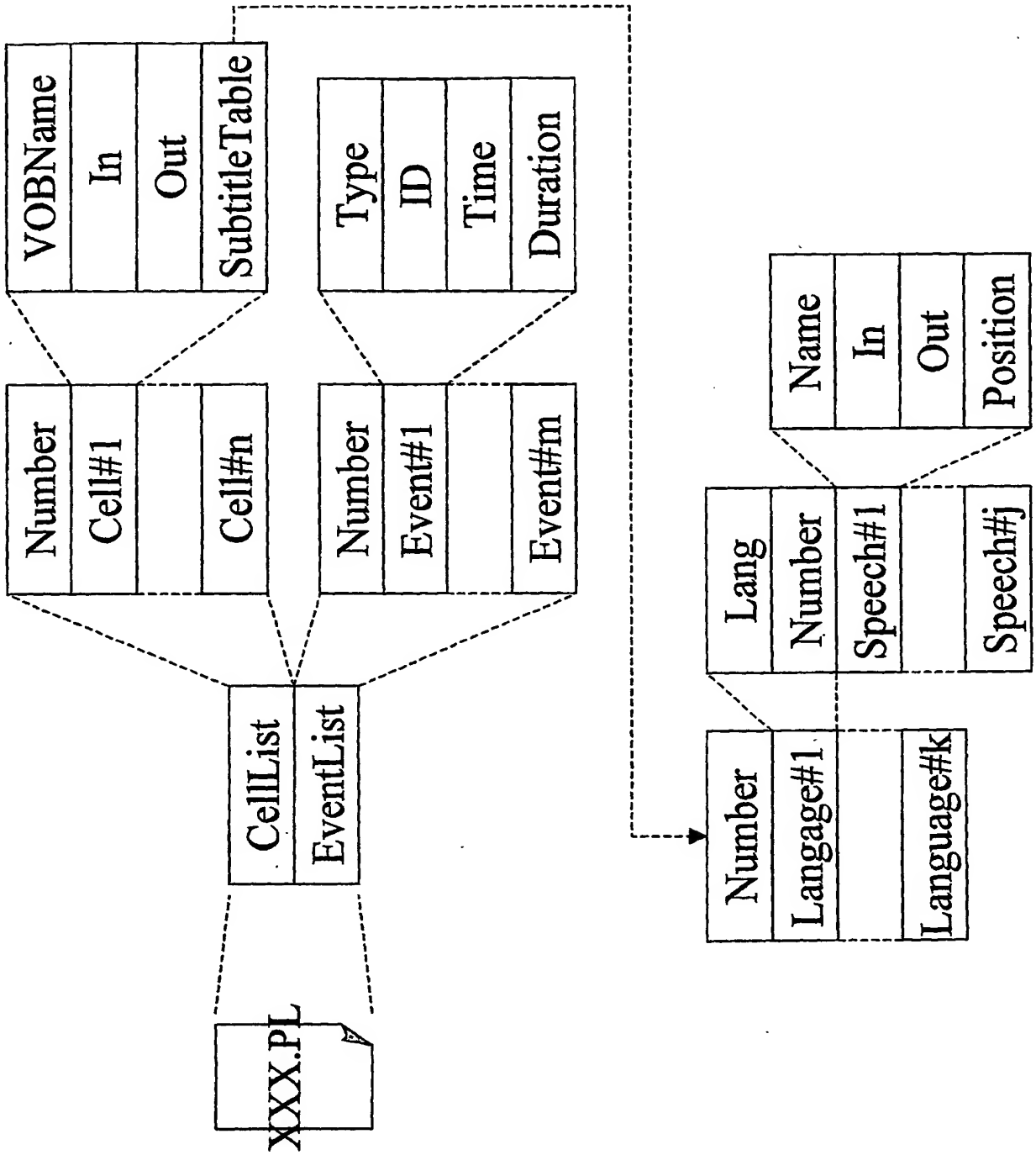
【図１４】


12frames
52packs
15frames
62packs
9frames
48packs
12frames
53packs
Duration
Size
タイムドメイン
アドレスドメイン

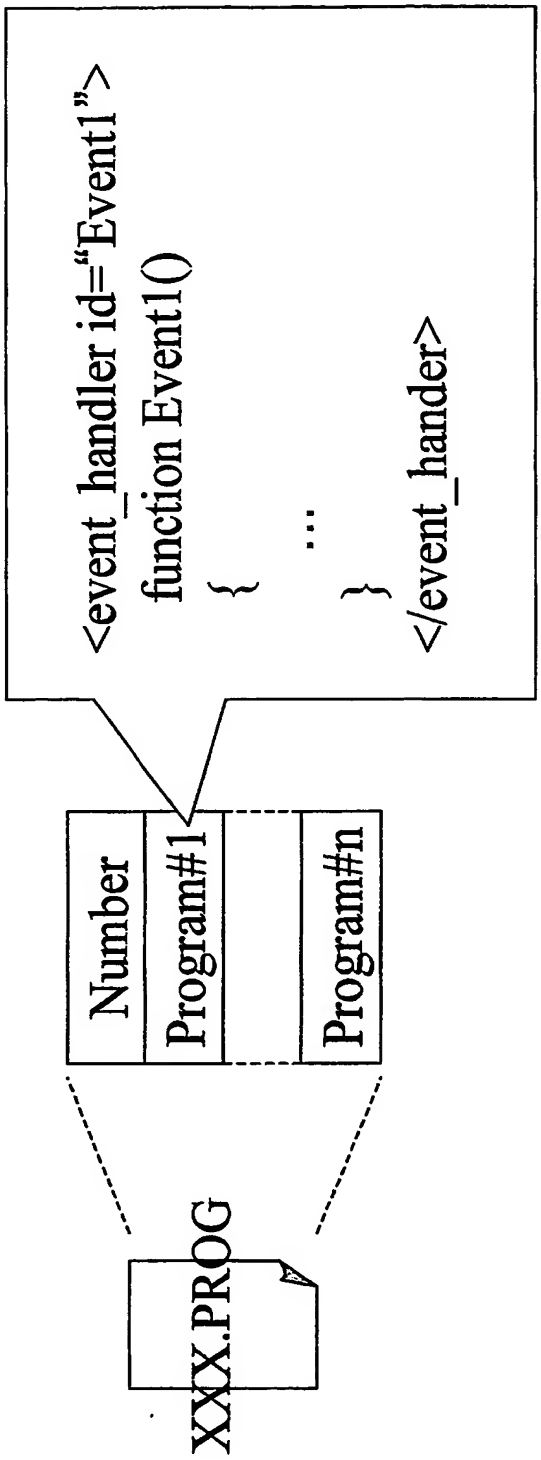
【図１５】



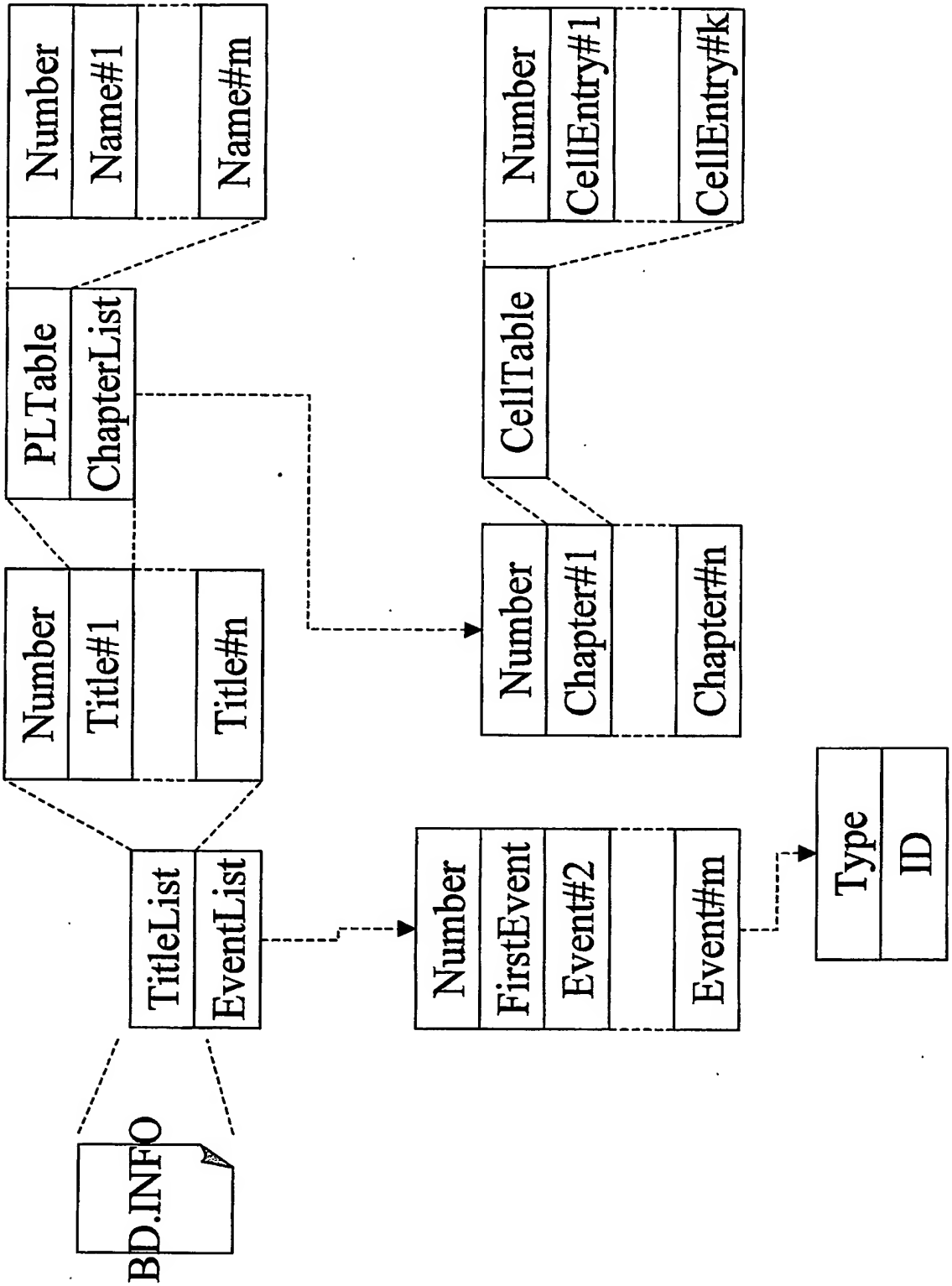
【図１６】


XXX.PL
CellList
EventList
Number
Cell#1
Cell#n
VOBName
In
Out
SubtitleTable
Number
Event#1
Event#m
Type
ID
Time
Duration
Number
Langage#1
Language#k
Lang
Number
Speech#1
Speech#j
Name
In
Out
Position

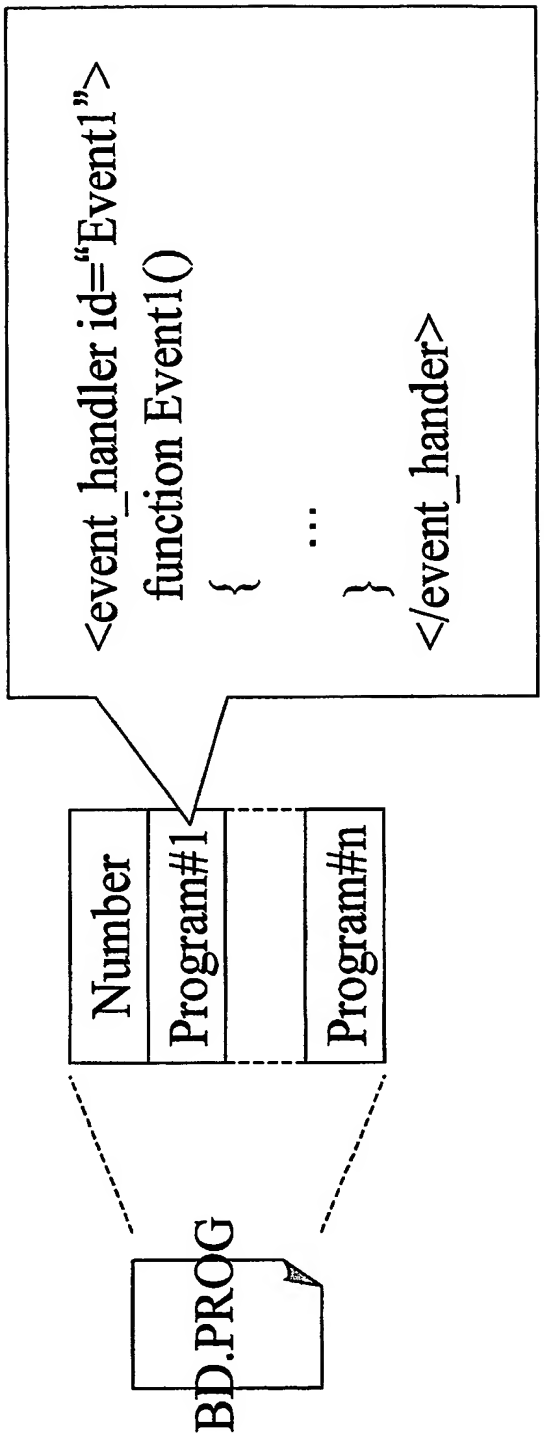
【図１７】


<event_handler id="Event1">
function Event1()
{
...
}
</event_hander>
Number
Program#1
Program#n
XXX.PROG

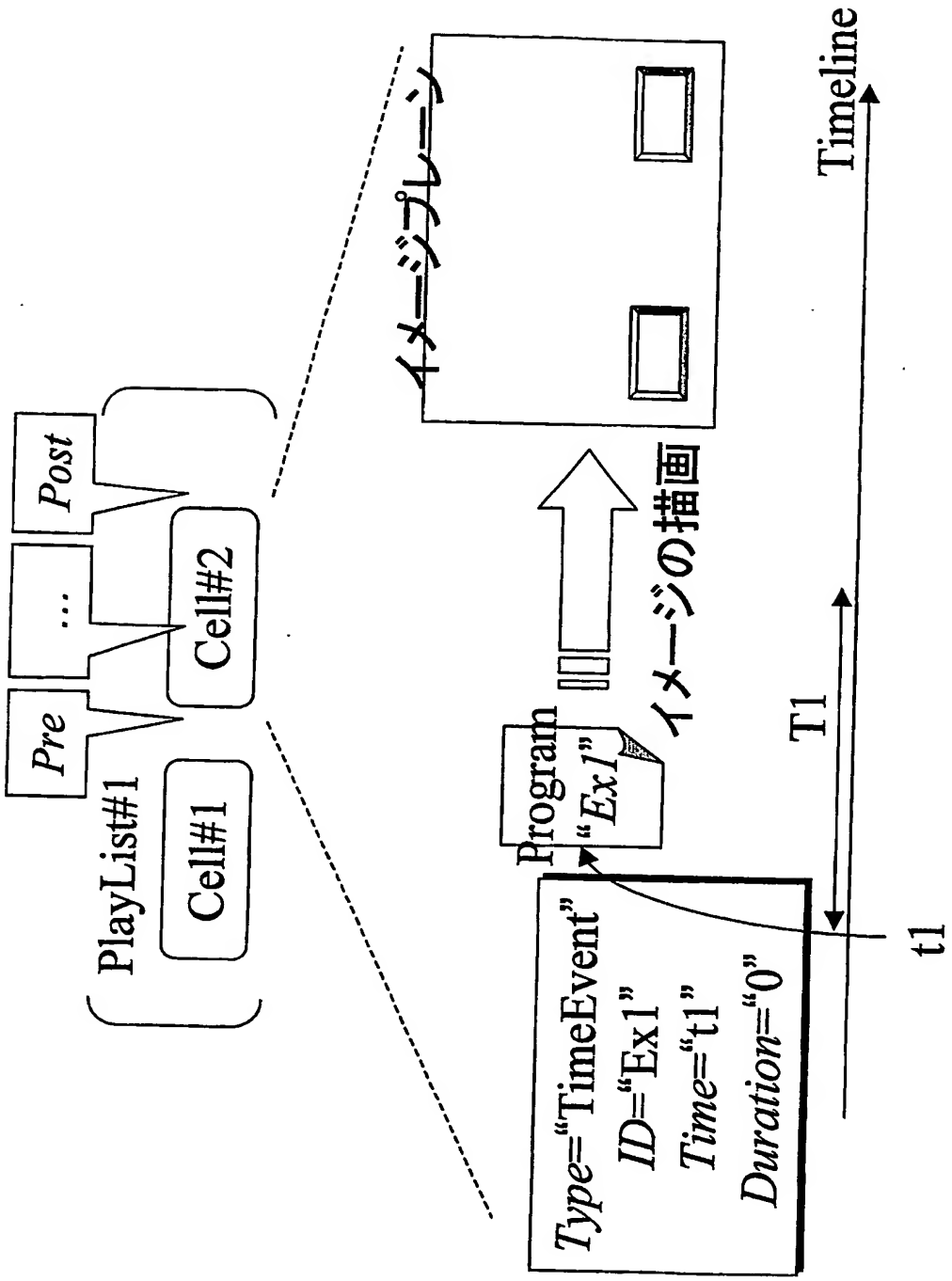
【図１８】


Number
Name#1
Name#m
PLTable
ChapterList
Number
Title#1
Title#n
TitleList
EventList
BD.INFO
Number
CellEntry#1
CellEntry#k
CellTable
Number
Chapter#1
Chapter#n
Number
FirstEvent
Event#2
Event#m
Type
ID

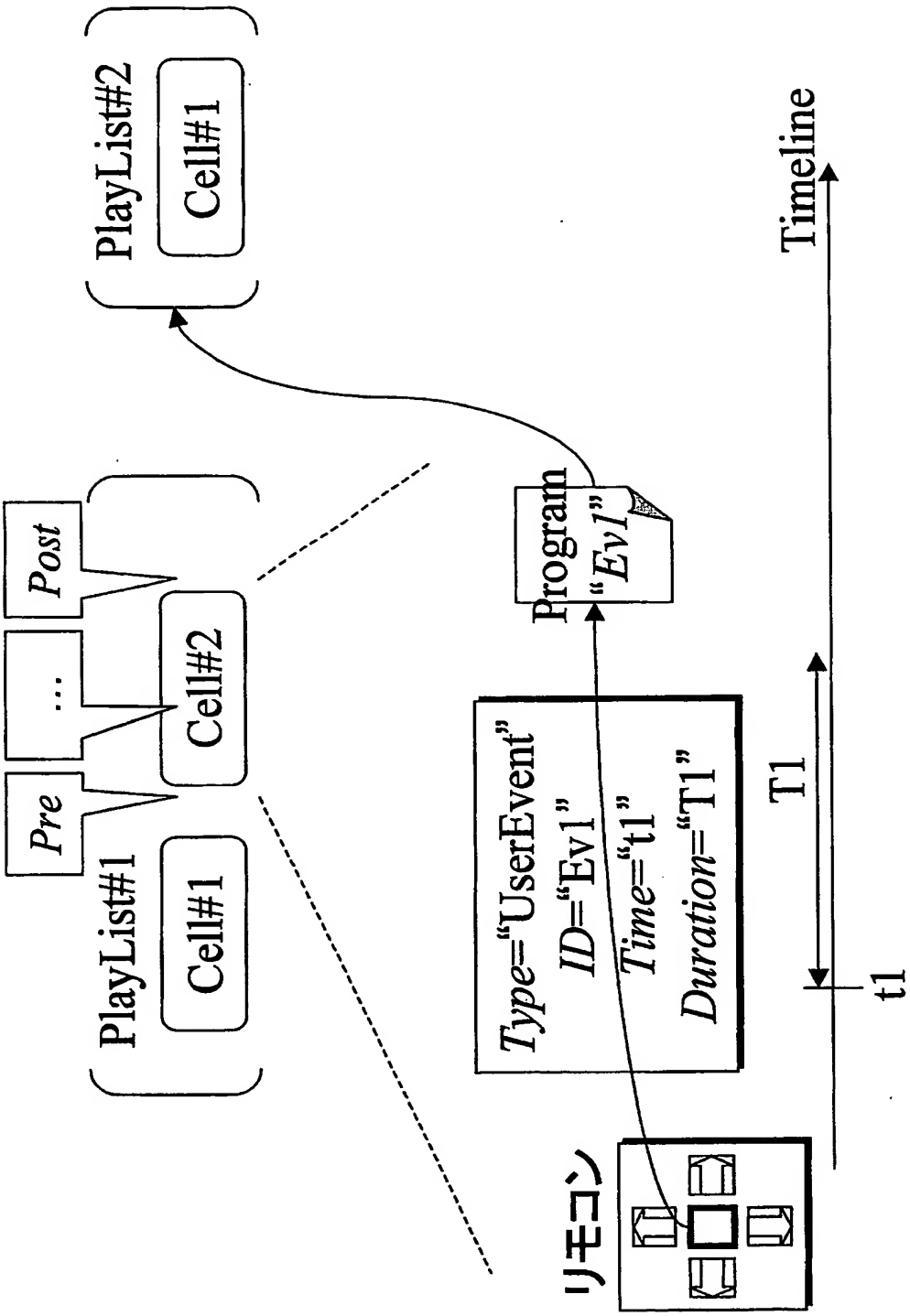
【図１９】


<event_handler id="Event1">
function Event1()
{
...
}
</event_hander>
Number
Program#1
Program#n
BD.PROG

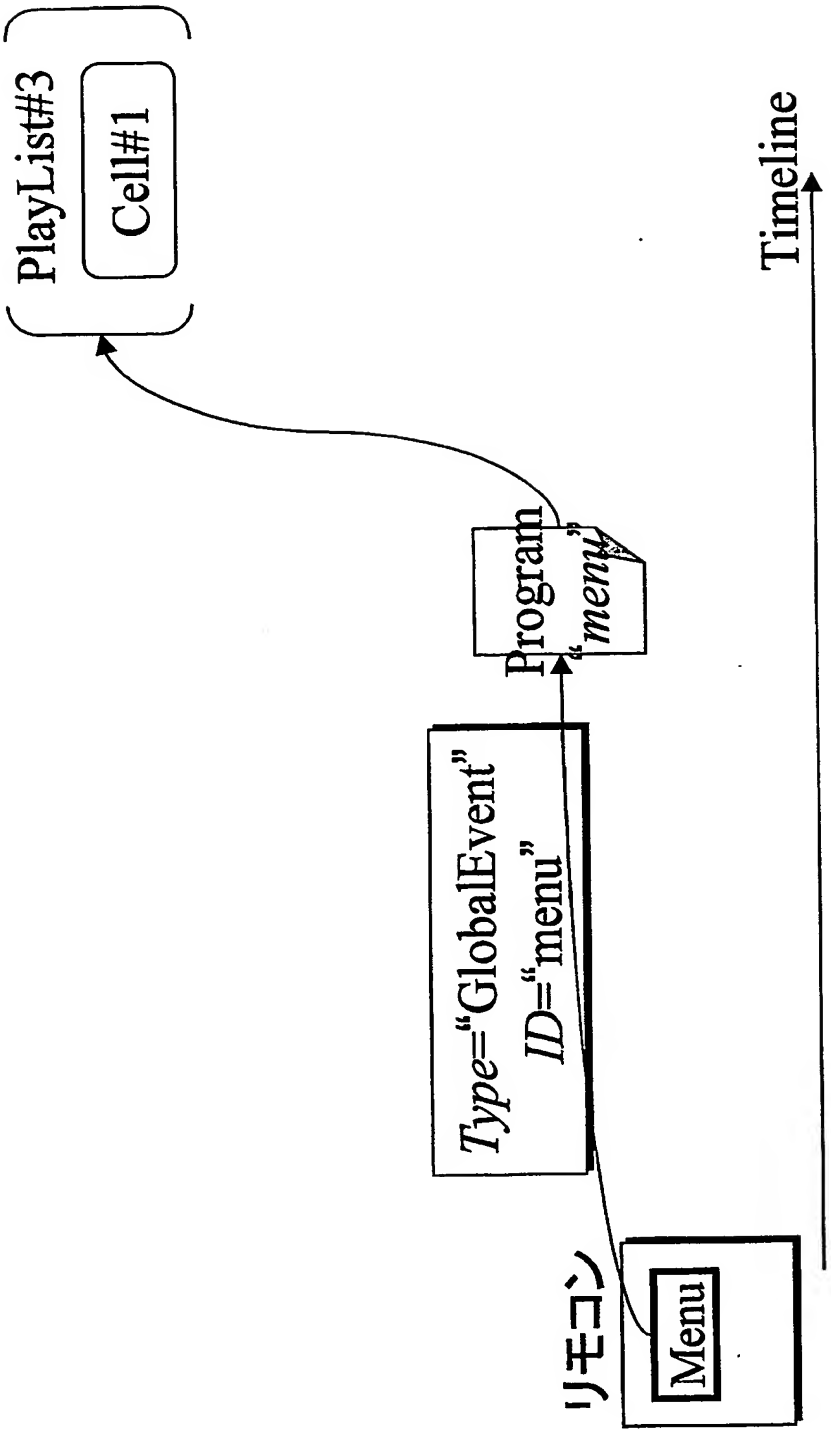
【図２０】


PlayList#1
Pre
Cell#1
...
Cell#2
Post
イメージプレーン
イメージの描画
Program
"Ex1"
Type="TimeEvent"
ID="Ex1"
Time="t1"
Duration="0"
T1
t1
Timeline

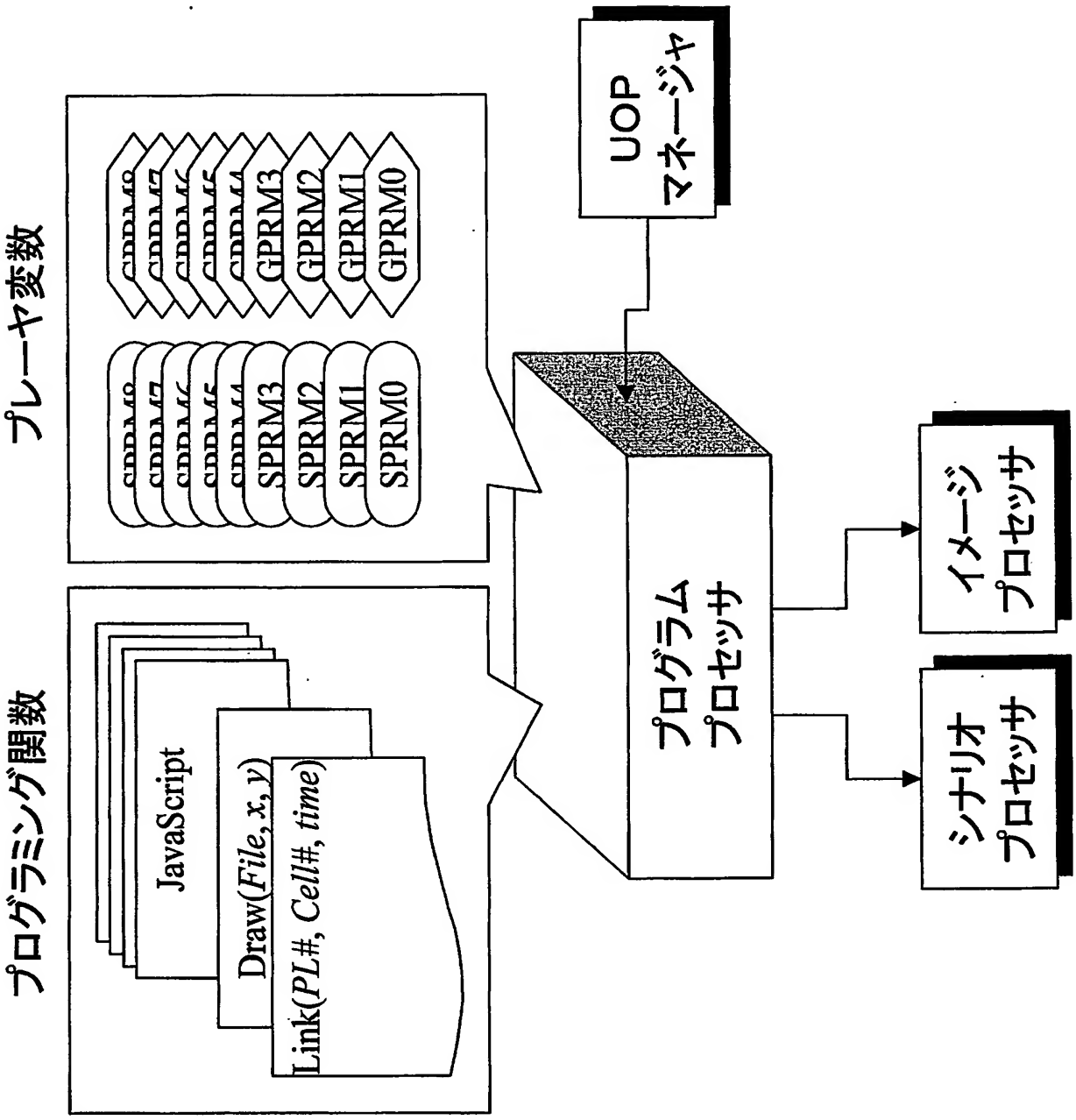
【図２１】


PlayList#2
Cell#1
Timeline
Post
...
Pre
Cell#2
PlayList#1
Cell#1
Program
"Ev1"
Type="UserEvent"
ID="Ev1"
Time="t1"
Duration="T1"
T1
t1
リモコン

【図２２】


PlayList#3
Cell#1
Timeline
Program
"menu"
Type="GlobalEvent"
ID="menu"
リモコン
Menu

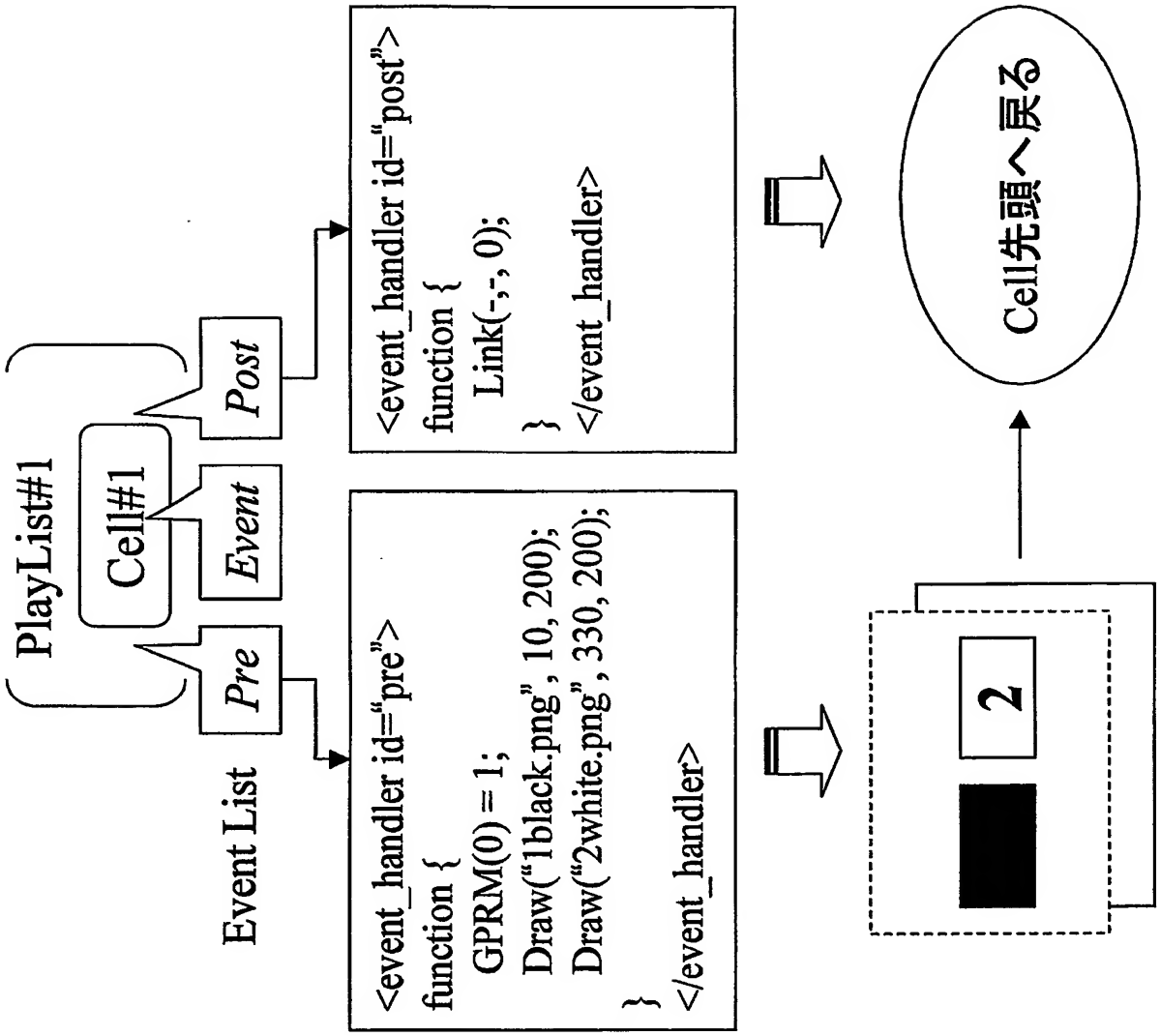
【図２３】


プレーヤ変数
GPRM3
GPRM2
GPRM1
GPRM0
SPRM3
SPRM2
SPRM1
SPRM0
UOP
マネージャ
プログラミング関数
JavaScript
Draw(File, x, y)
Link(PL#, Cell#, time)
プログラム
プロセッサ
イメージ
プロセッサ
シナリオ
プロセッサ

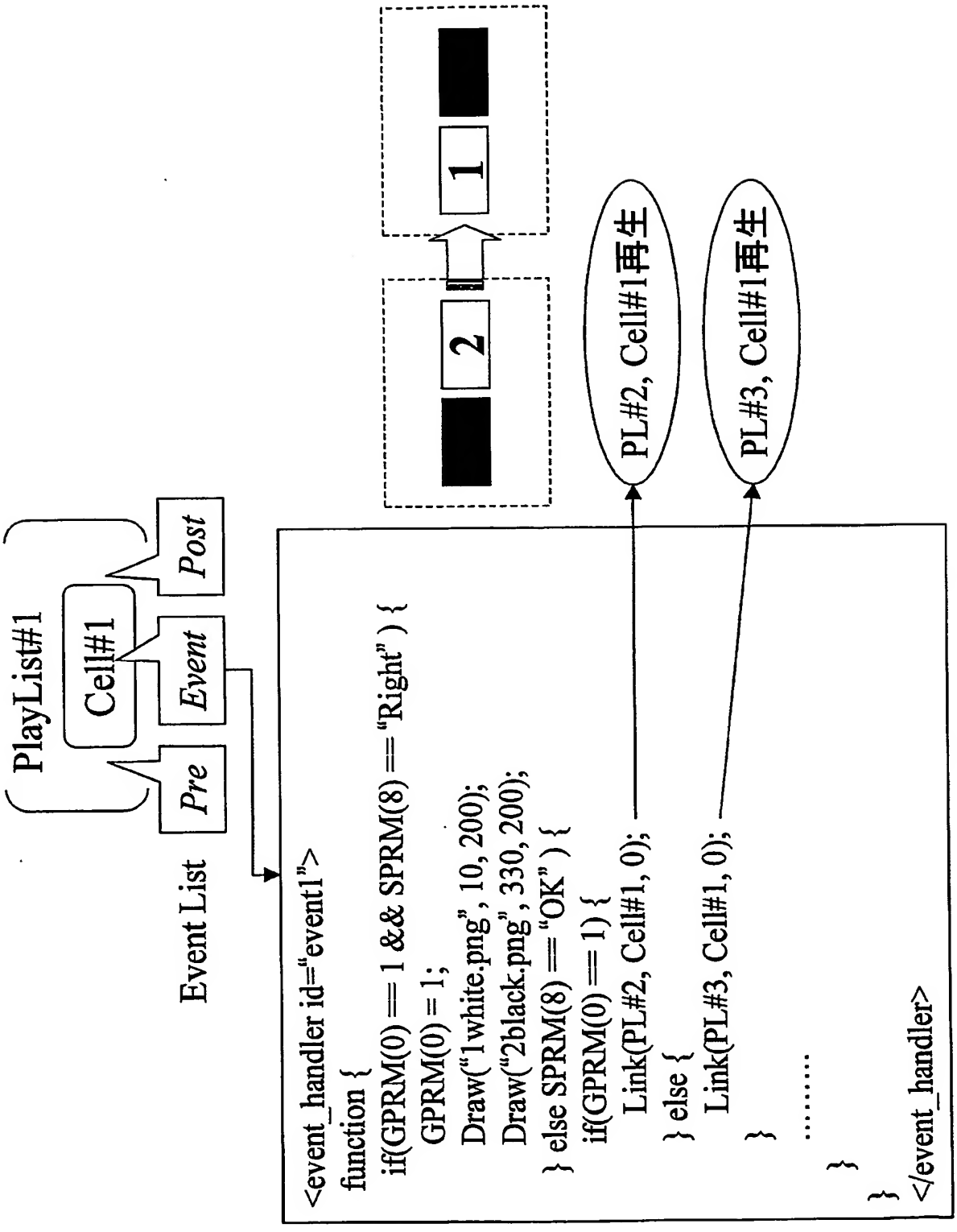
プレーヤ変数(システムパラメータ)

0	Language Code	11	Player audio mixing mode for Karaoke	22	reserved
1	Audio stream number	12	Country code for parental management	23	Player status
2	Subtitle stream number	13	Parental level	24	reserved
3	Angle number	14	Player configuration for Video	25	reserved
4	Title number	15	Player configuration for Audio	26	reserved
5	Chapter number	16	Language code for AST	27	reserved
6	Program number	17	Language code ext. for AST	28	reserved
7	Cell number	18	Language code for STST	29	reserved
8	Key name	19	Language coded ext. for STST	30	reserved
9	Navigation timer	20	Player region code	31	reserved
10	Current playback time	21	reserved	32	reserved

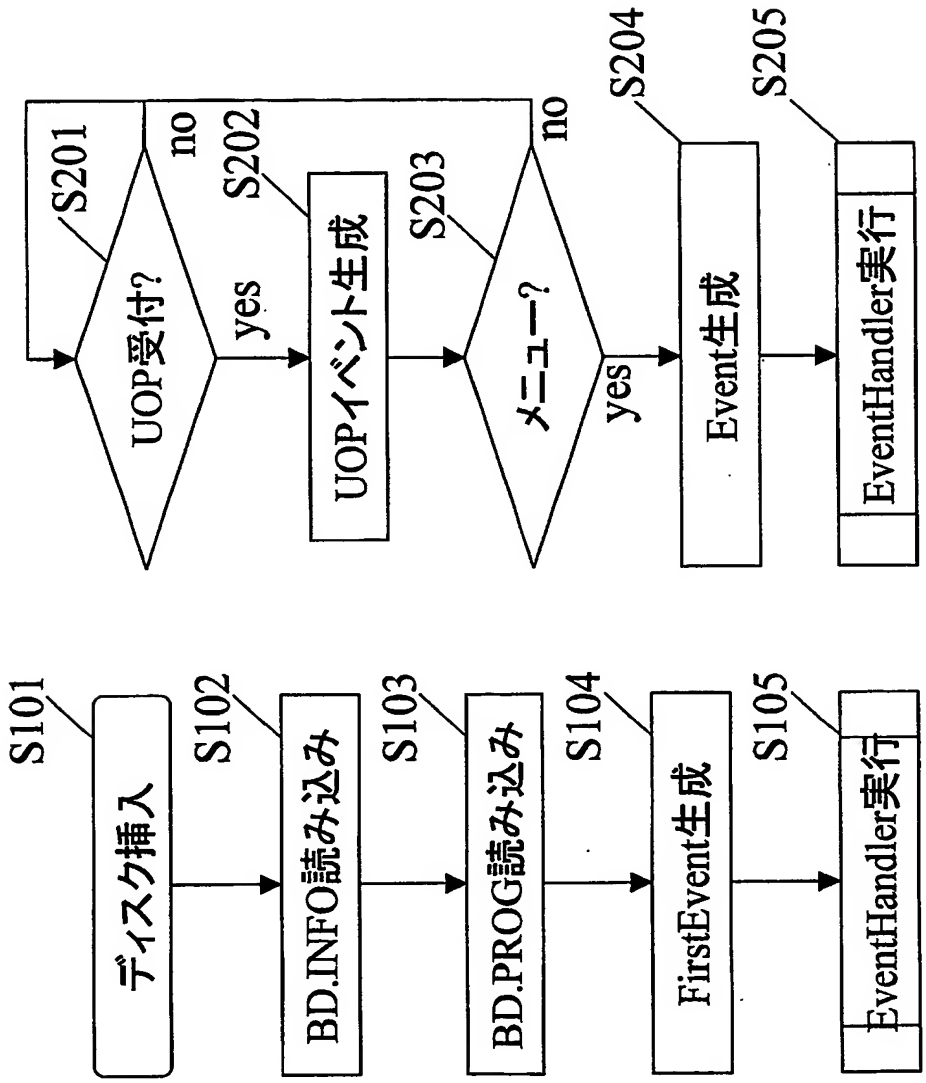
【図２５】


PlayList#1
Cell#1
Pre
Event
Post
Event List
<event_handler id="pre">
function {
GPRM(0) = 1;
Draw("1black.png", 10, 200);
Draw("2white.png", 330, 200);
}
</event_handler>
<event_handler id="post">
function {
Link(-,-, 0);
}
</event_handler>
2
Cell先頭へ戻る

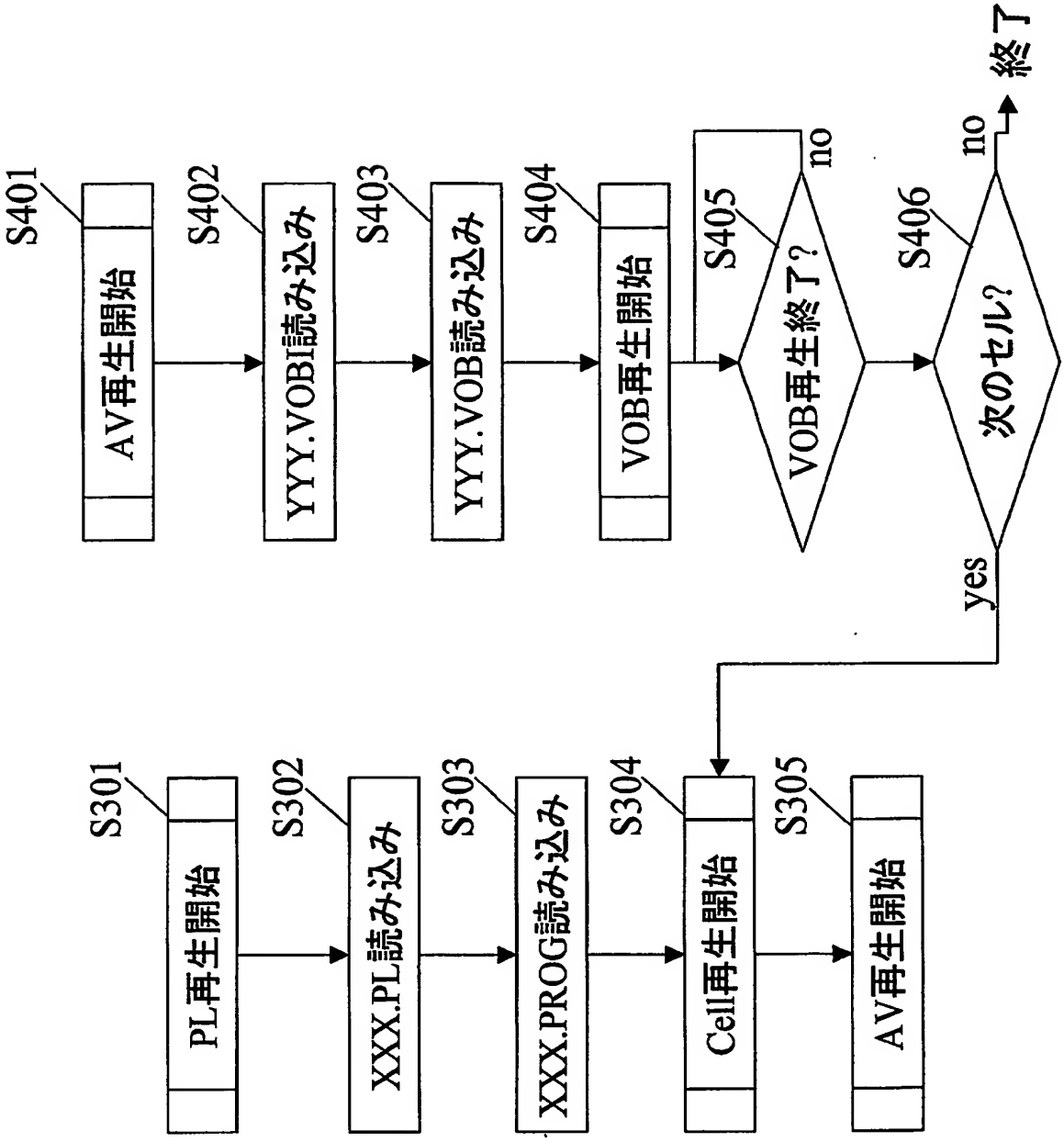
【図２６】


PlayList#1
Cell#1
Pre
Event
Post
Event List
<event_handler id="event1">
function {
if(GPRM(0) == 1 && SPRM(8) == "Right") {
GPRM(0) = 1;
Draw("1white.png", 10, 200);
Draw("2black.png", 330, 200);
} else SPRM(8) == "OK") {
if(GPRM(0) == 1) {
Link(PL#2, Cell#1, 0);
} else {
Link(PL#3, Cell#1, 0);
}
.....
}
}
</event_handler>
2
1
PL#2, Cell#1再生
PL#3, Cell#1再生

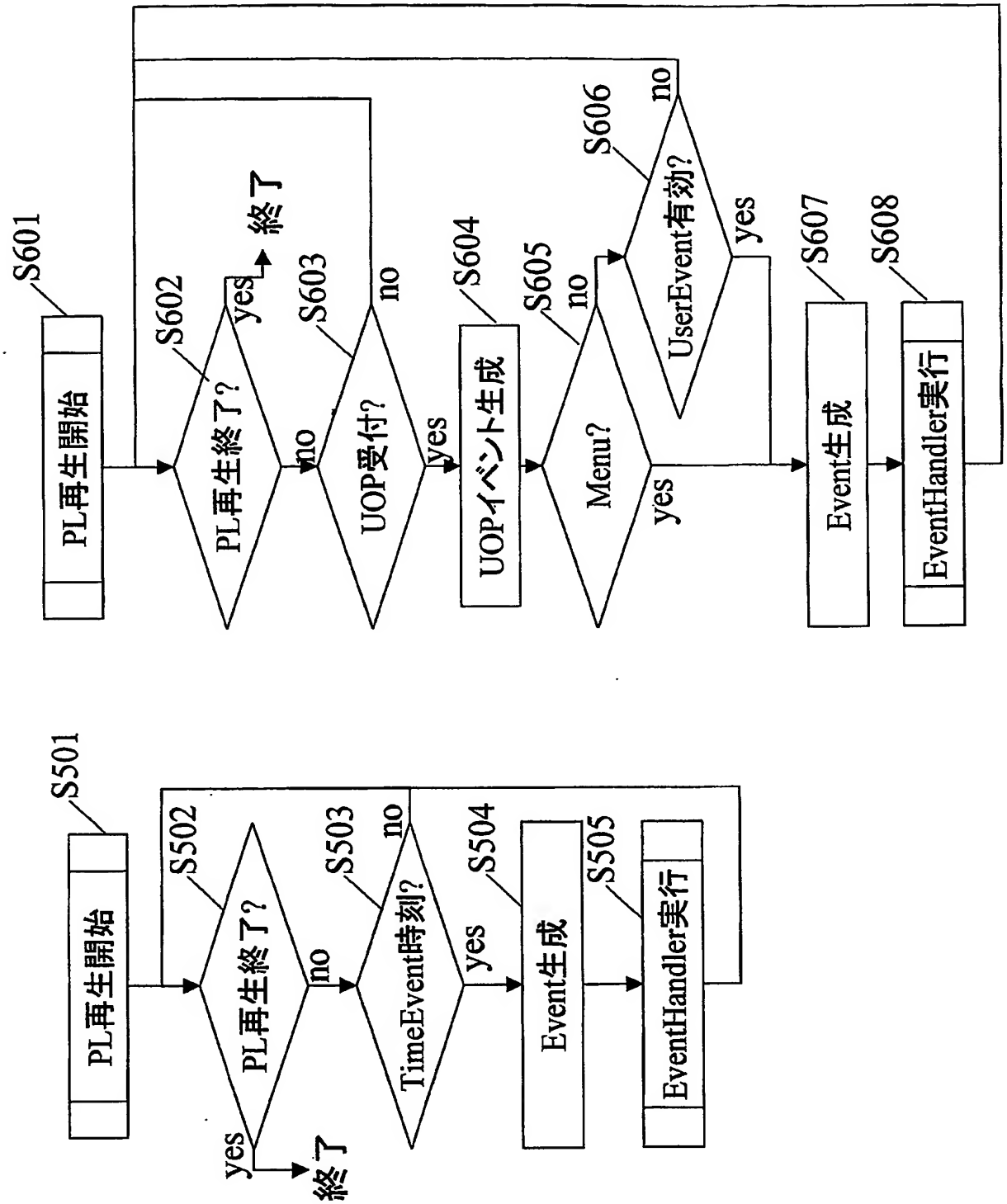
【図２７】


S101
ディスク挿入
S102
BD.INFO読み込み
S103
BD.PROG読み込み
S104
FirstEvent生成
S105
EventHandler実行
S201
UOP受付?
no
yes
S202
UOPイベント生成
S203
メニュー?
no
yes
S204
Event生成
S205
EventHandler実行

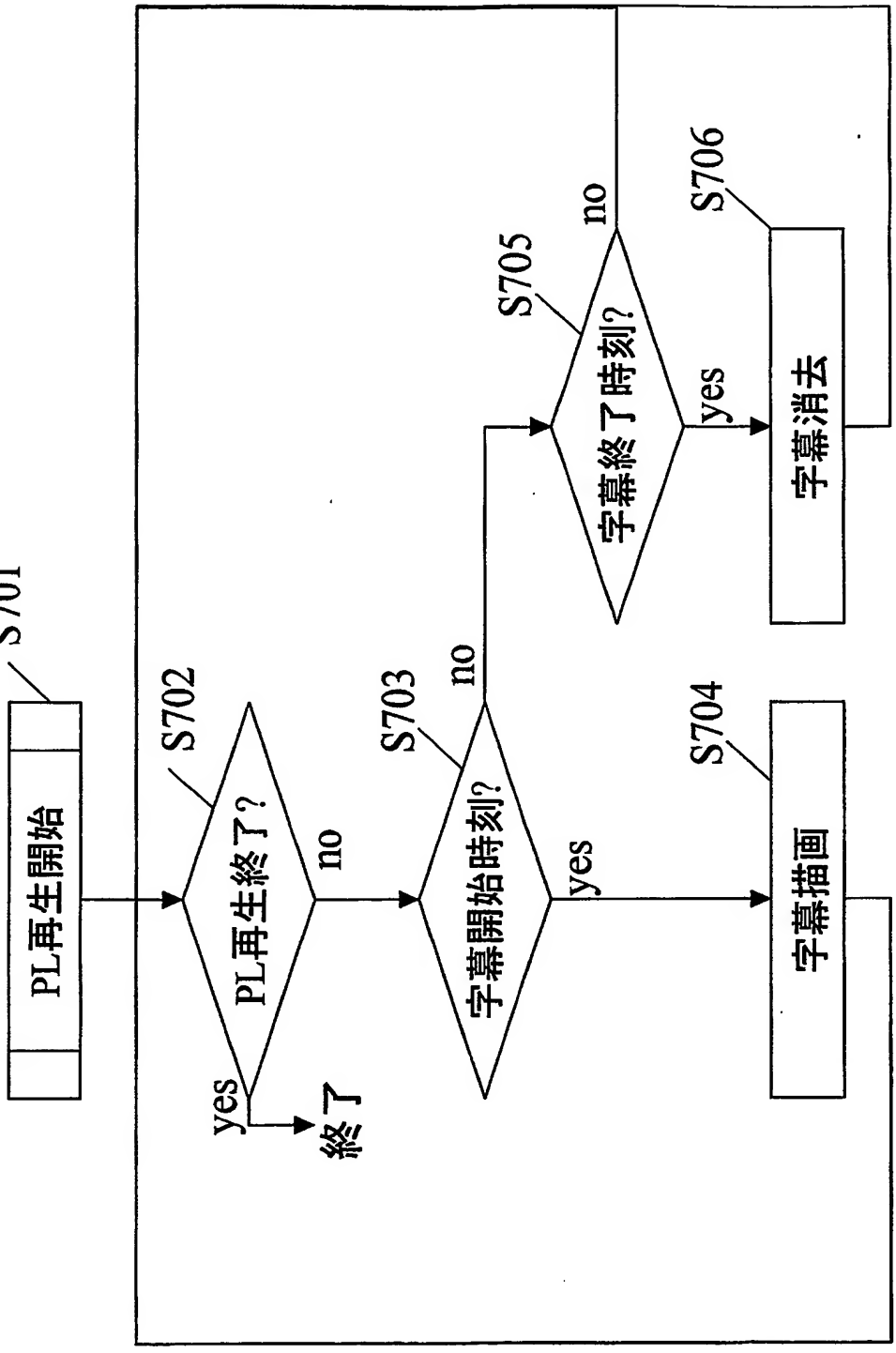
【図２８】


S401
AV再生開始
S402
YYY.VOBI読み込み
S403
YYY.VOB読み込み
S404
VOB再生開始
S405
VOB再生終了?
no
S406
次のセル?
no
終了
yes
S301
PL再生開始
S302
XXX.PL読み込み
S303
XXX.PROG読み込み
S304
Cell再生開始
S305
AV再生開始

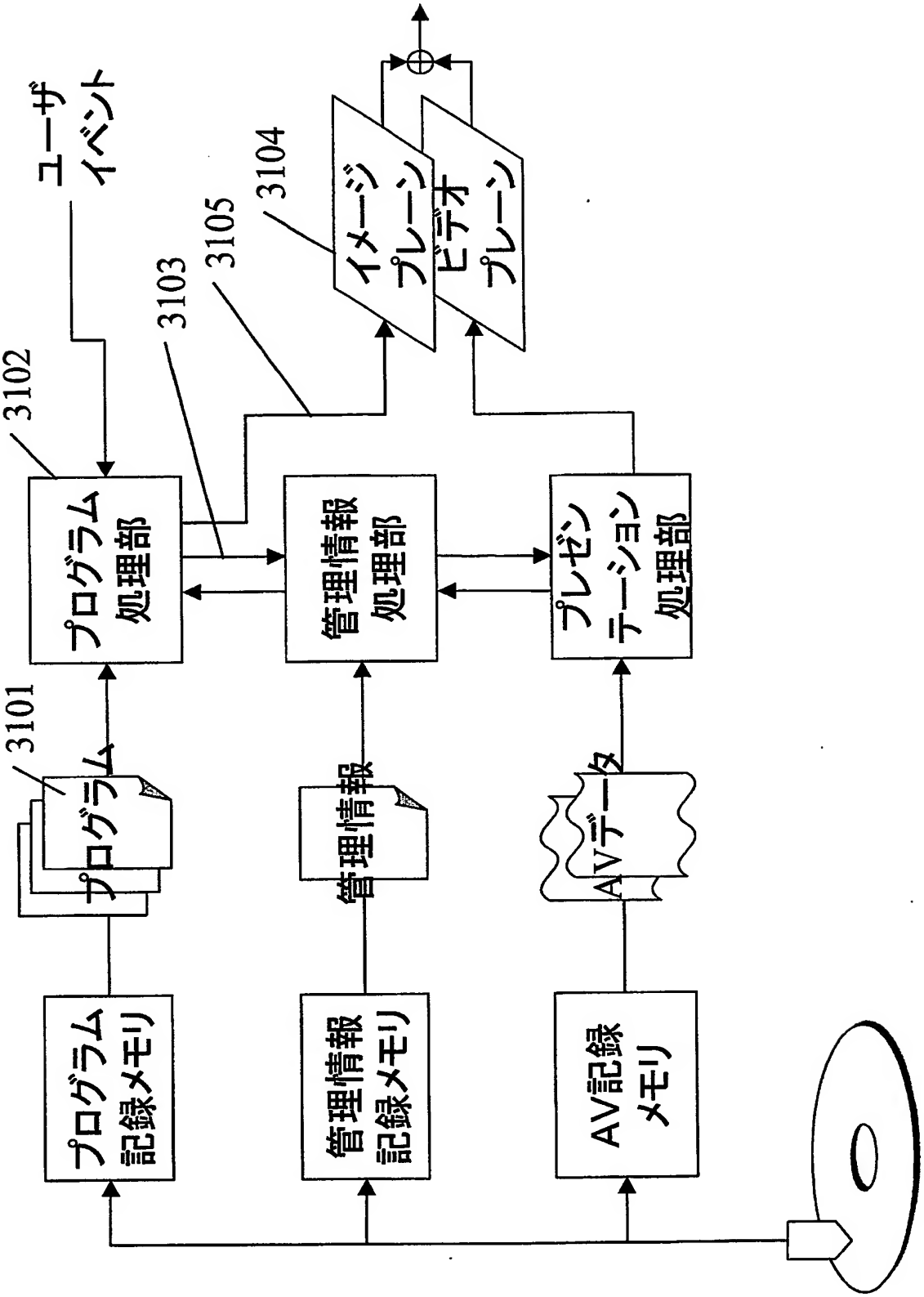
【図２９】


S601
PL再生開始
S602
PL再生終了?
yes
終了
no
S603
UOP受付?
no
yes
S604
UOPイベント生成
S605
Menu?
no
yes
S606
UserEvent有効?
no
yes
S607
Event生成
S608
EventHandler実行
S501
PL再生開始
S502
PL再生終了?
yes
終了
no
S503
TimeEvent時刻?
no
yes
S504
Event生成
S505
EventHandler実行

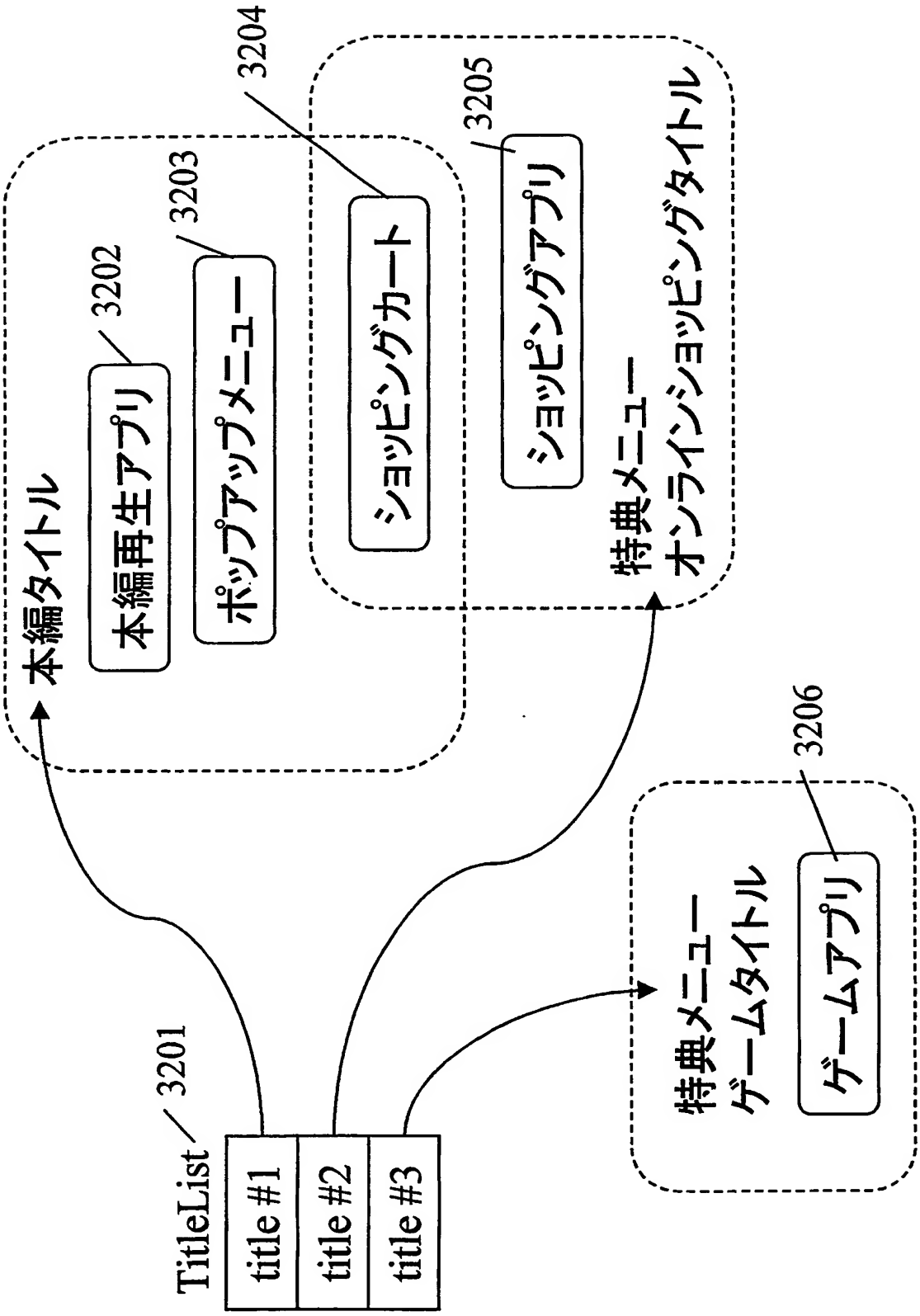
【図３０】


S701
PL再生開始
S702
PL再生終了?
yes
終了
no
S703
字幕開始時刻?
yes
S704
字幕描画
no
S705
字幕終了時刻?
yes
S706
字幕消去
no

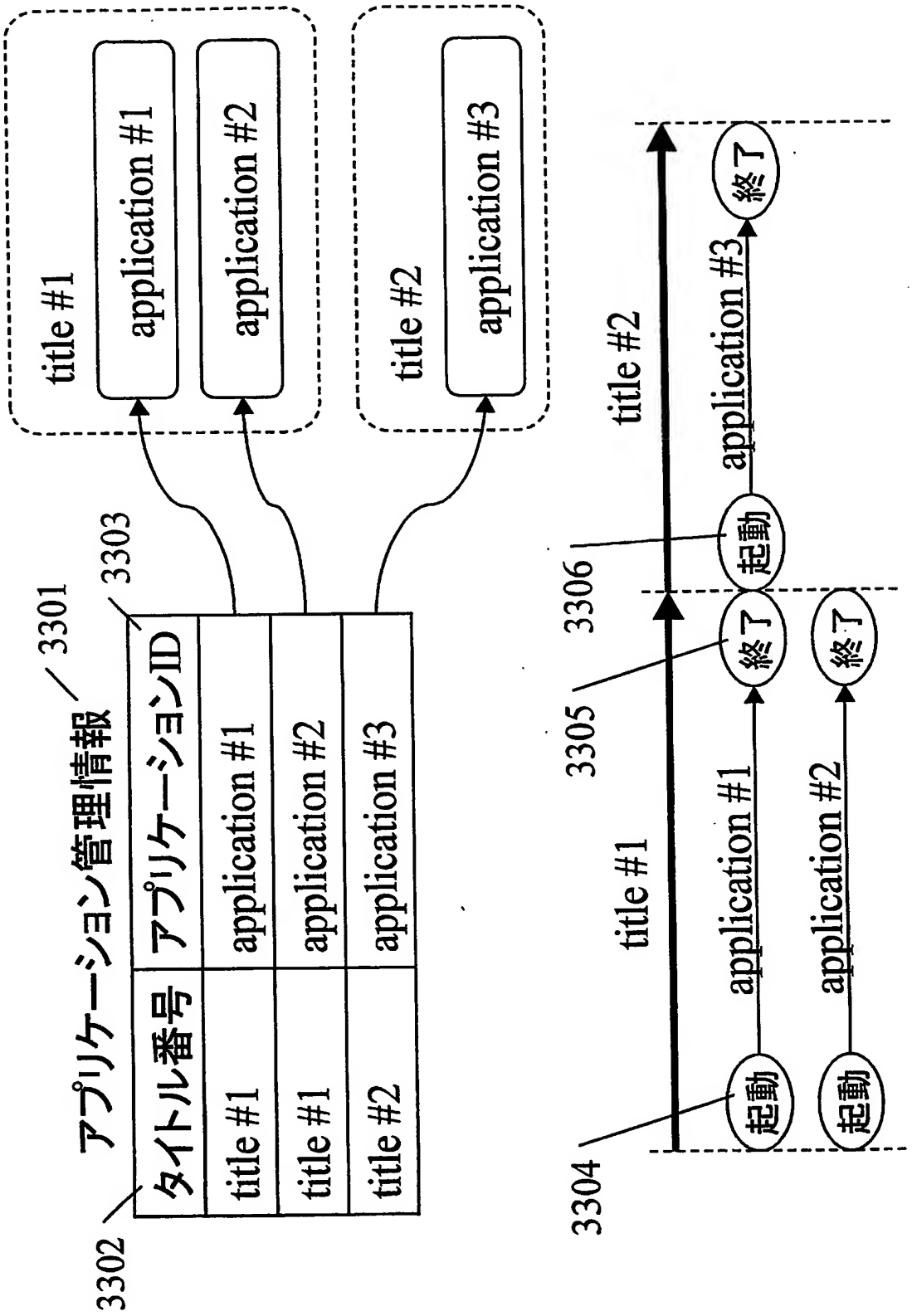
【図３１】


ユーザ
イベント
3104
3105
3103
3102
3101
イメージ
プレーン
ビデオ
プレーン
プログラム
処理部
管理情報
処理部
プレゼン
テーション
処理部
プログラム
管理情報
AVデータ
プログラム
記録メモリ
管理情報
記録メモリ
AV記録
メモリ

【図３２】


TitleList
3201
title #1
title #2
title #3
本編タイトル
3202
本編再生アプリ
3203
ポップアップメニュー
3204
ショッピングカート
3205
ショッピングアプリ
特典メニュー
オンラインショッピングタイトル
特典メニュー
ゲームタイトル
3206
ゲームアプリ

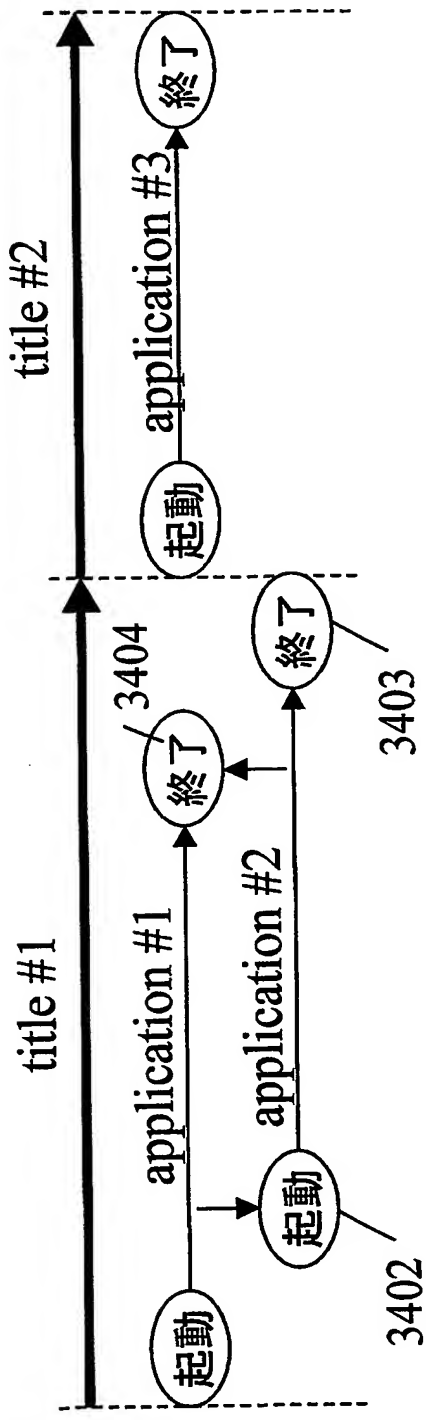
【図３３】


title #1
application #1
application #2
title #2
application #3
アプリケーション管理情報
3301
3303
タイトル番号
アプリケーションID
title #1
application #1
title #1
application #2
title #2
application #3
3302
3304
3305
3306
起動
終了
起動
終了
起動
終了

【図３４】

アプリケーション管理情報　3401

タイトル番号	アプリケーションID	起動属性
title #1	application #1	AutoRun
title #1	application #2	-
title #2	application #3	AutoRun

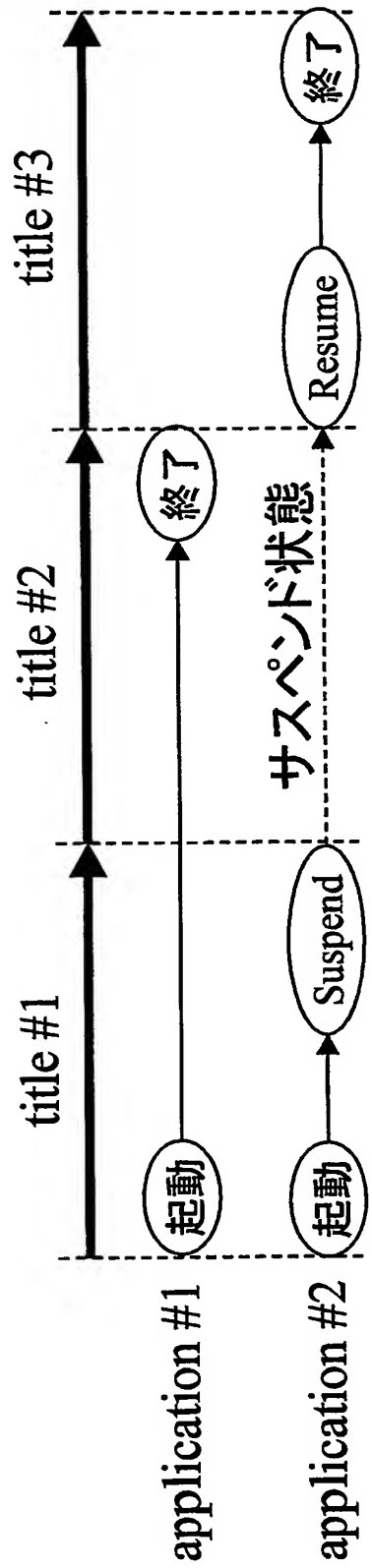


【図３５】

タイトル番号	アプリケーションID	起動属性
title #1	application #1	AutoRun
title #1	application #2	AutoRun
title #2	application #1	-
title #2	application #2	Suspend
title #3	application #2	-

3501

3502



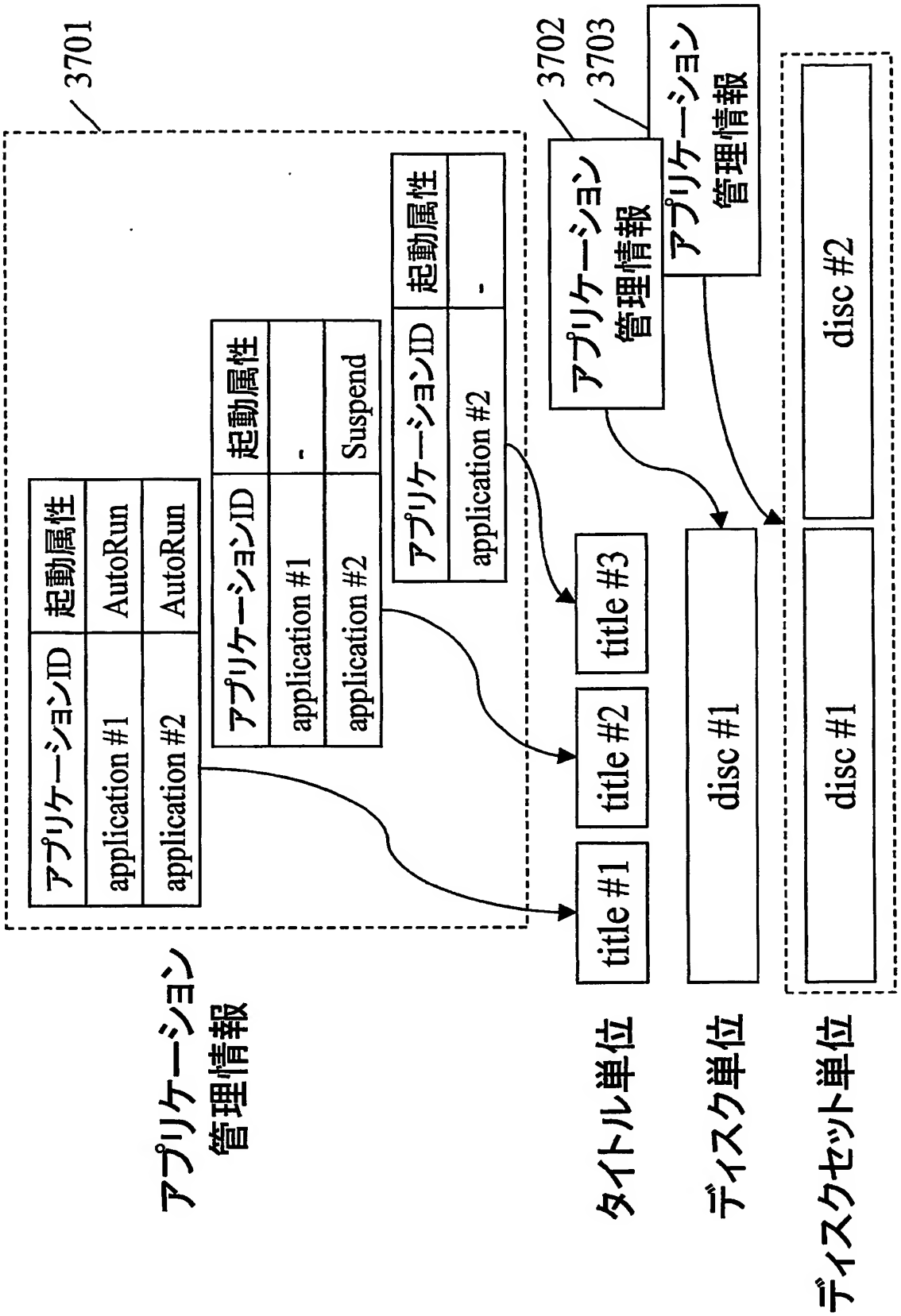
【図３６】

起動属性によるアプリケーション状態の変化

直前のタイトルにおけるアプリケーションの状態	起動属性 (指定なし)	起動属性 AutoRun	起動属性 Suspend
起動していない	-	アプリケーションを起動する	-
起動している	-	-	サスペンドする
サスペンドしている	レジュームする	レジュームする	-

"-"：何もせず、前の状態を継続する

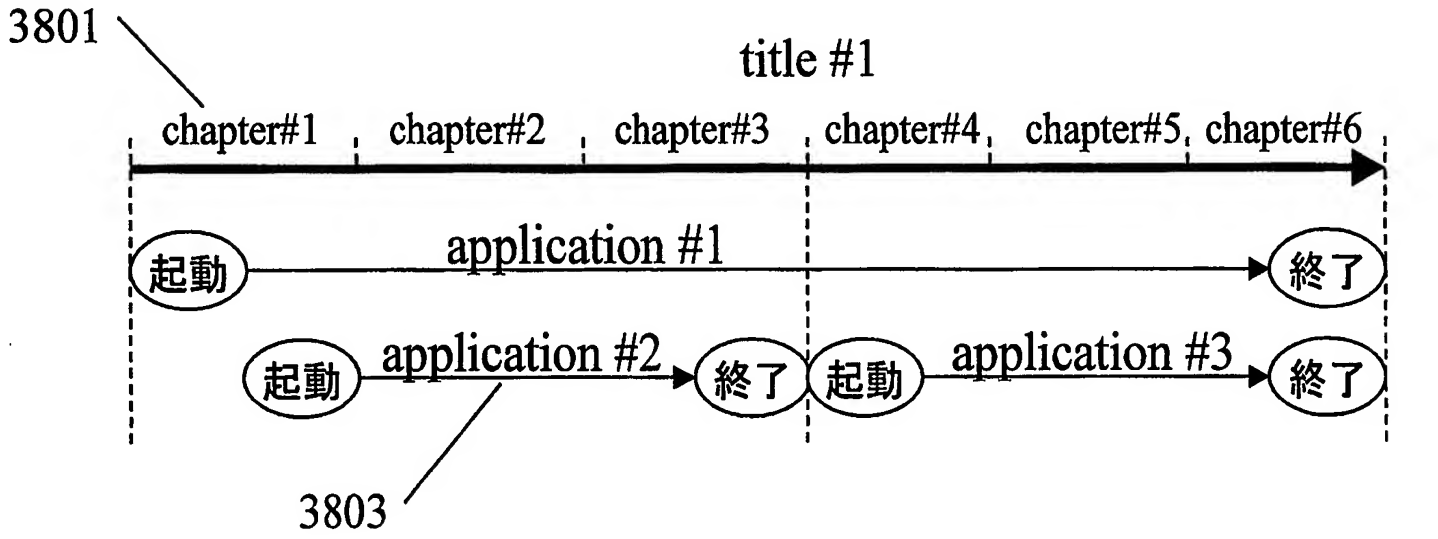
【図３７】


3701
アプリケーションID
起動属性
application #1
AutoRun
application #2
AutoRun
アプリケーションID
起動属性
application #1
-
application #2
Suspend
アプリケーションID
起動属性
application #2
-
3702
アプリケーション管理情報
3703
アプリケーション管理情報
title #1
title #2
title #3
disc #1
disc #1
disc #2
アプリケーション管理情報
タイトル単位
ディスク単位
ディスクセット単位

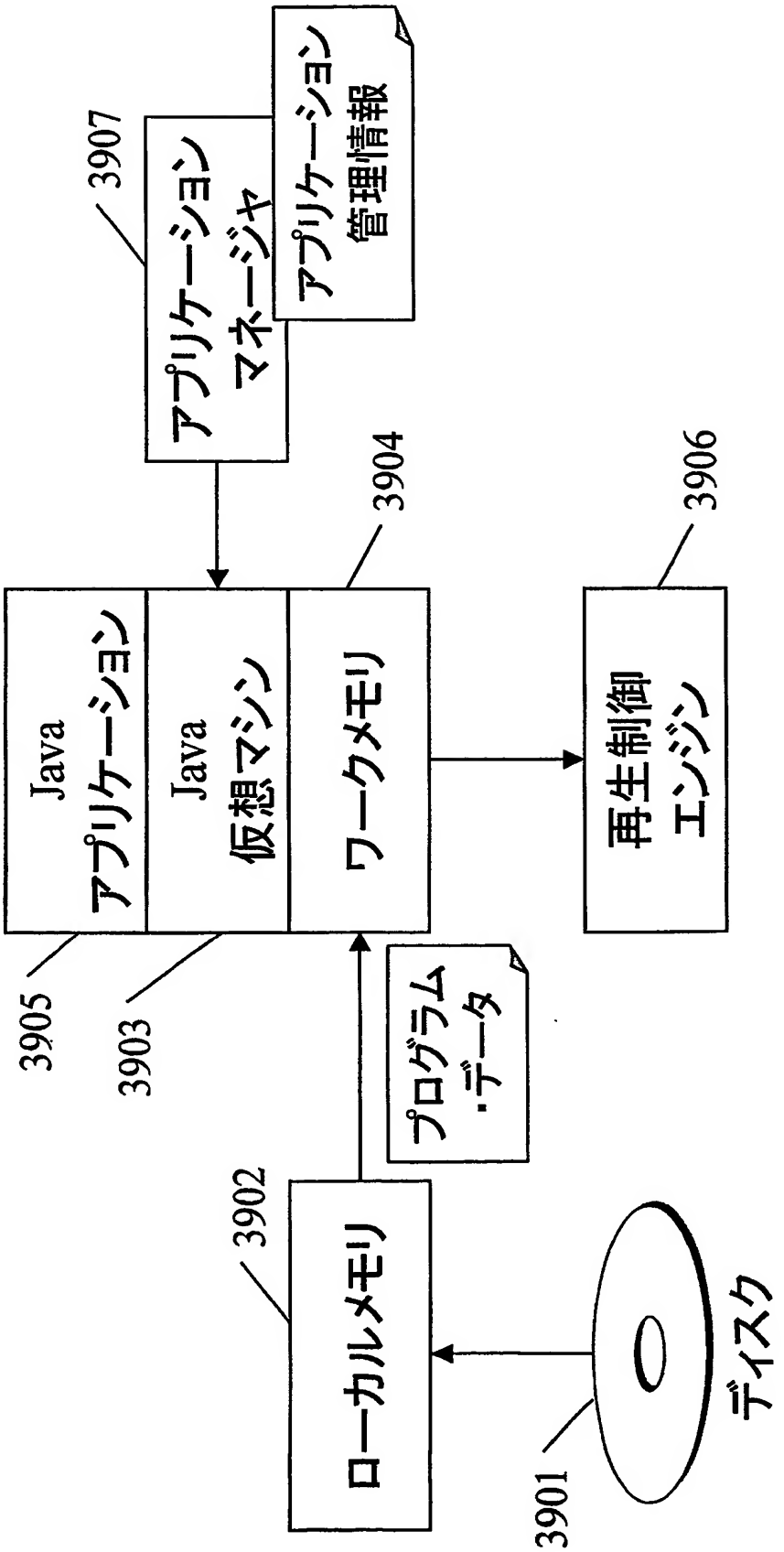
アプリケーション管理情報

有効期間	アプリケーションID	起動属性
title #1	application #1	AutoRun
title #1 : chapter #1 - #3	application #2	-
title #1 : chapter #4 - #6	application #3	AutoRun

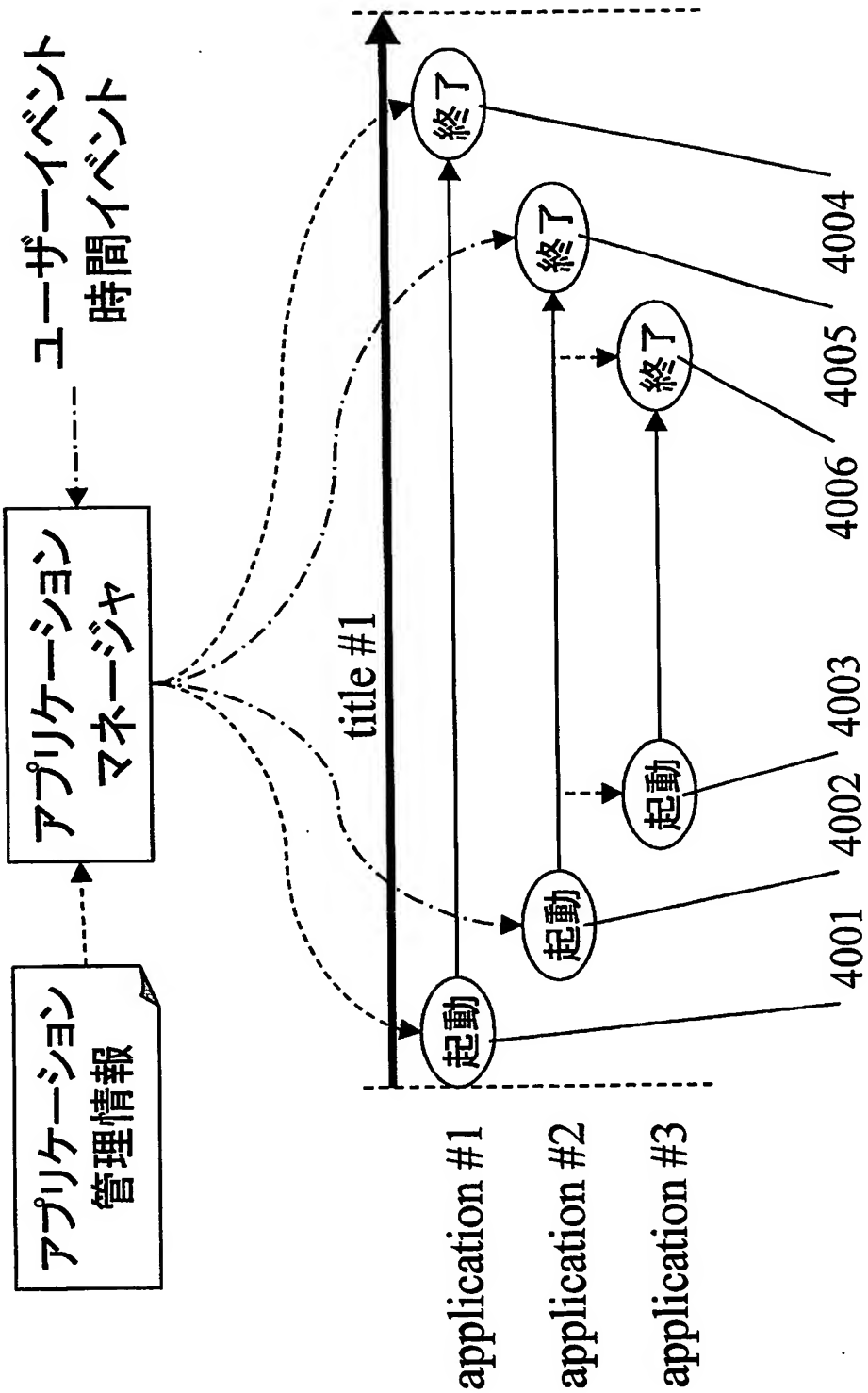
3802



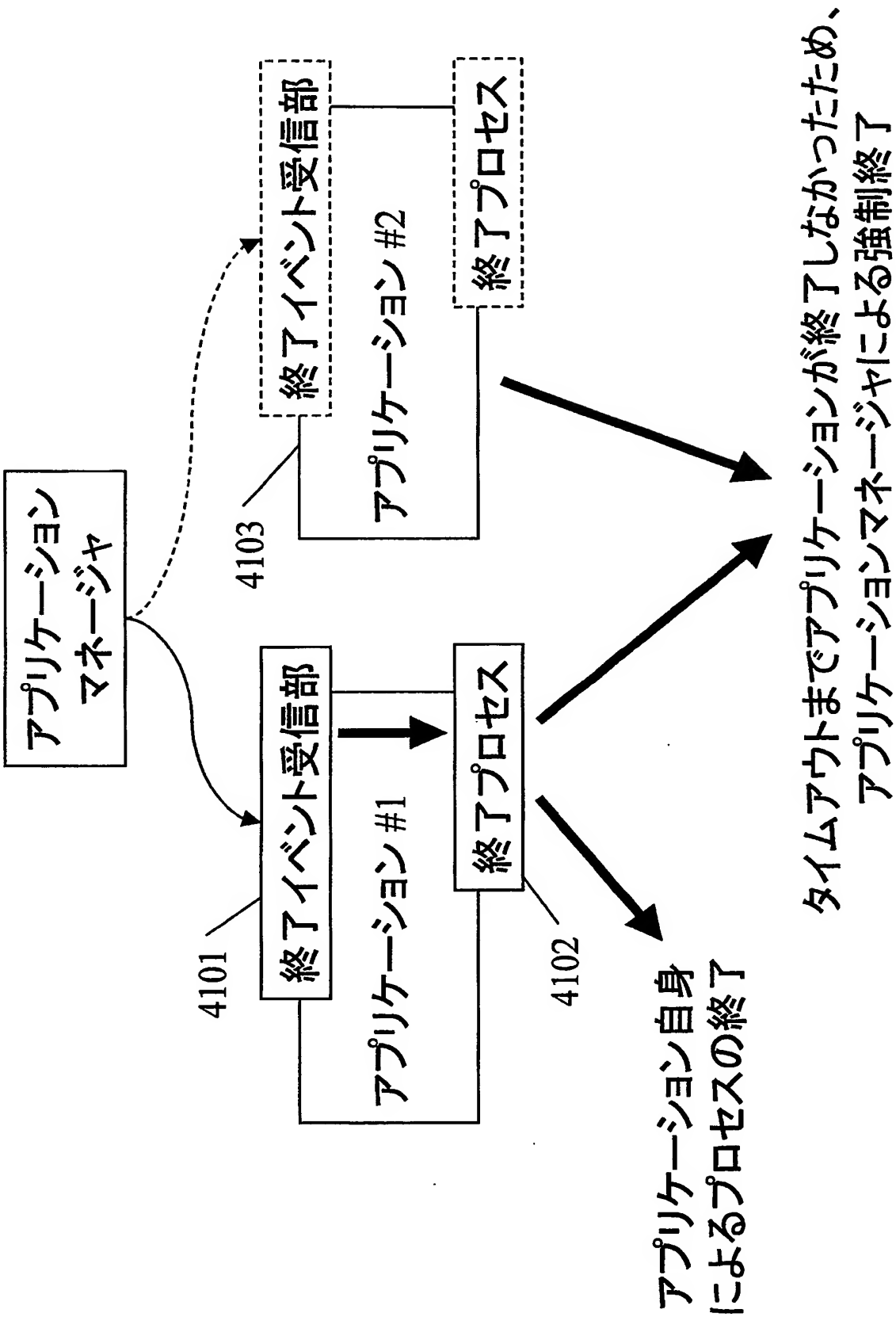
【図３９】


アプリケーション管理情報
3907
アプリケーションマネージャ
3904
3906
Java アプリケーション
Java 仮想マシン
ワークメモリ
再生制御エンジン
3905
3903
プログラム・データ
3902
ローカルメモリ
ディスク
3901

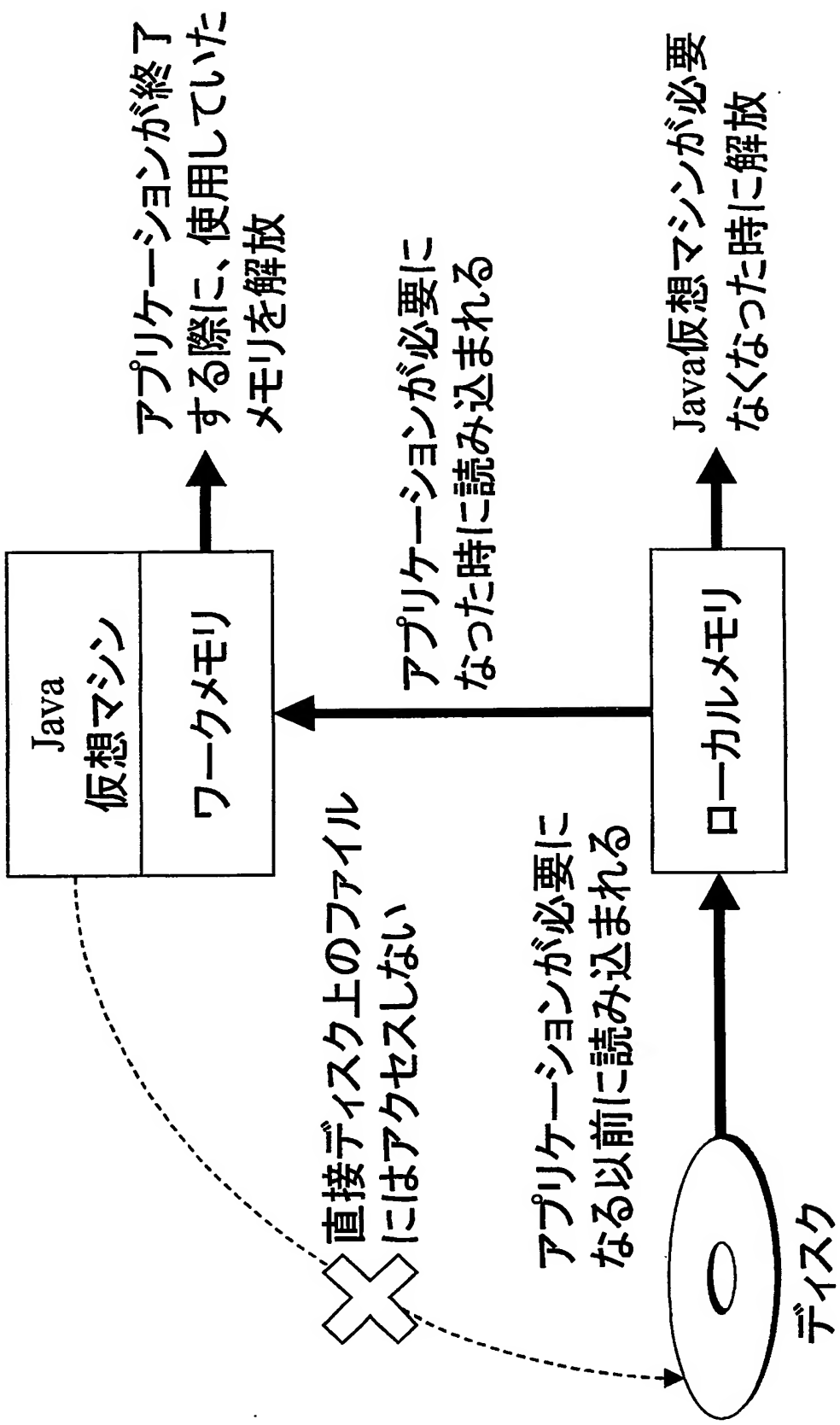
【図４０】


アプリケーション
管理情報
アプリケーション
マネージャ
ユーザーイベント
時間イベント
title #1
application #1
application #2
application #3
起動
起動
起動
終了
終了
終了
4001
4002
4003
4004
4005
4006

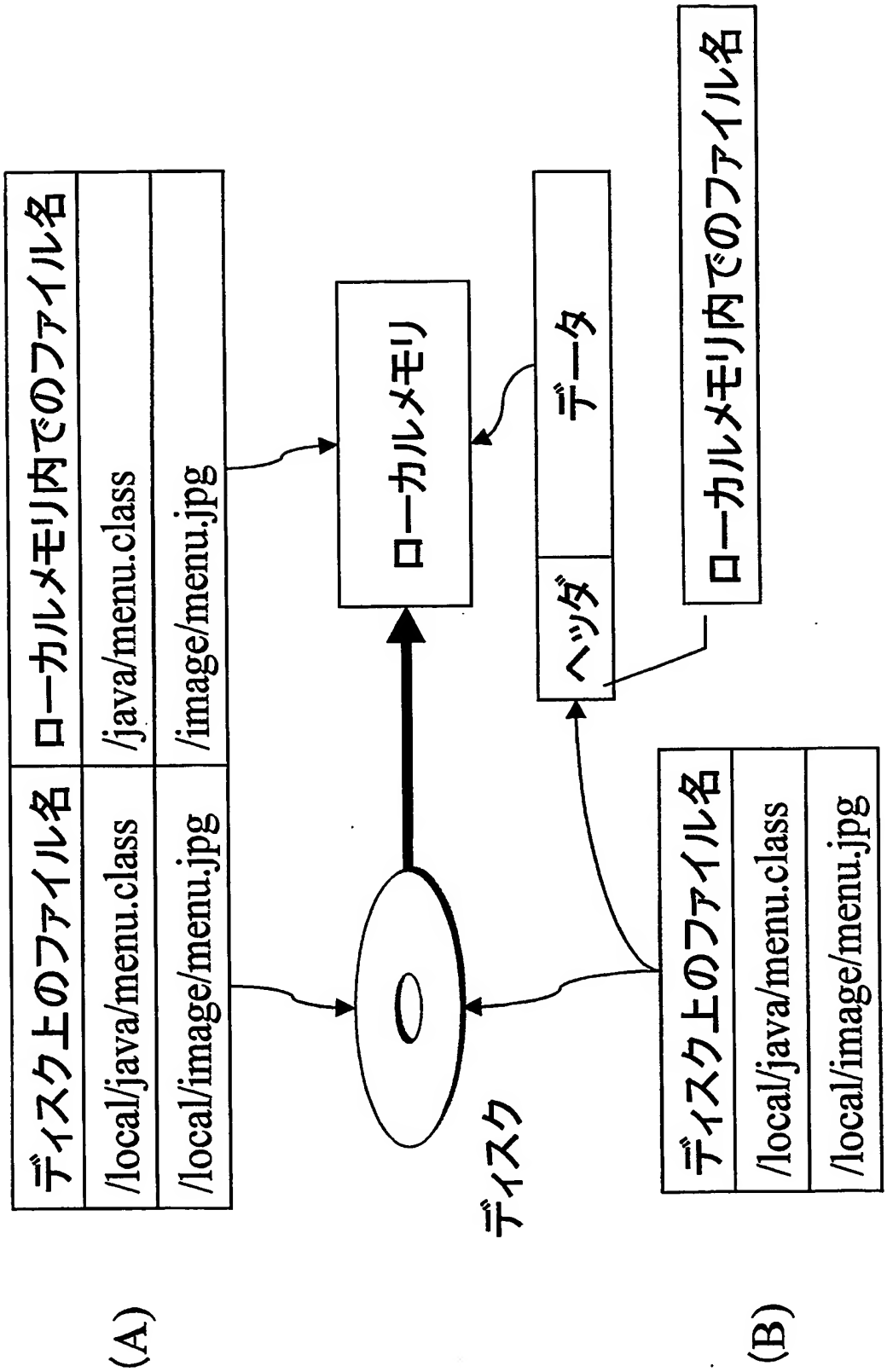
【図４１】


アプリケーション
マネージャ
終了イベント受信部
アプリケーション #1
終了プロセス
4101
4102
終了イベント受信部
アプリケーション #2
終了プロセス
4103
アプリケーション自身
によるプロセスの終了
タイムアウトまでアプリケーションが終了しなかったため、
アプリケーションマネージャによる強制終了

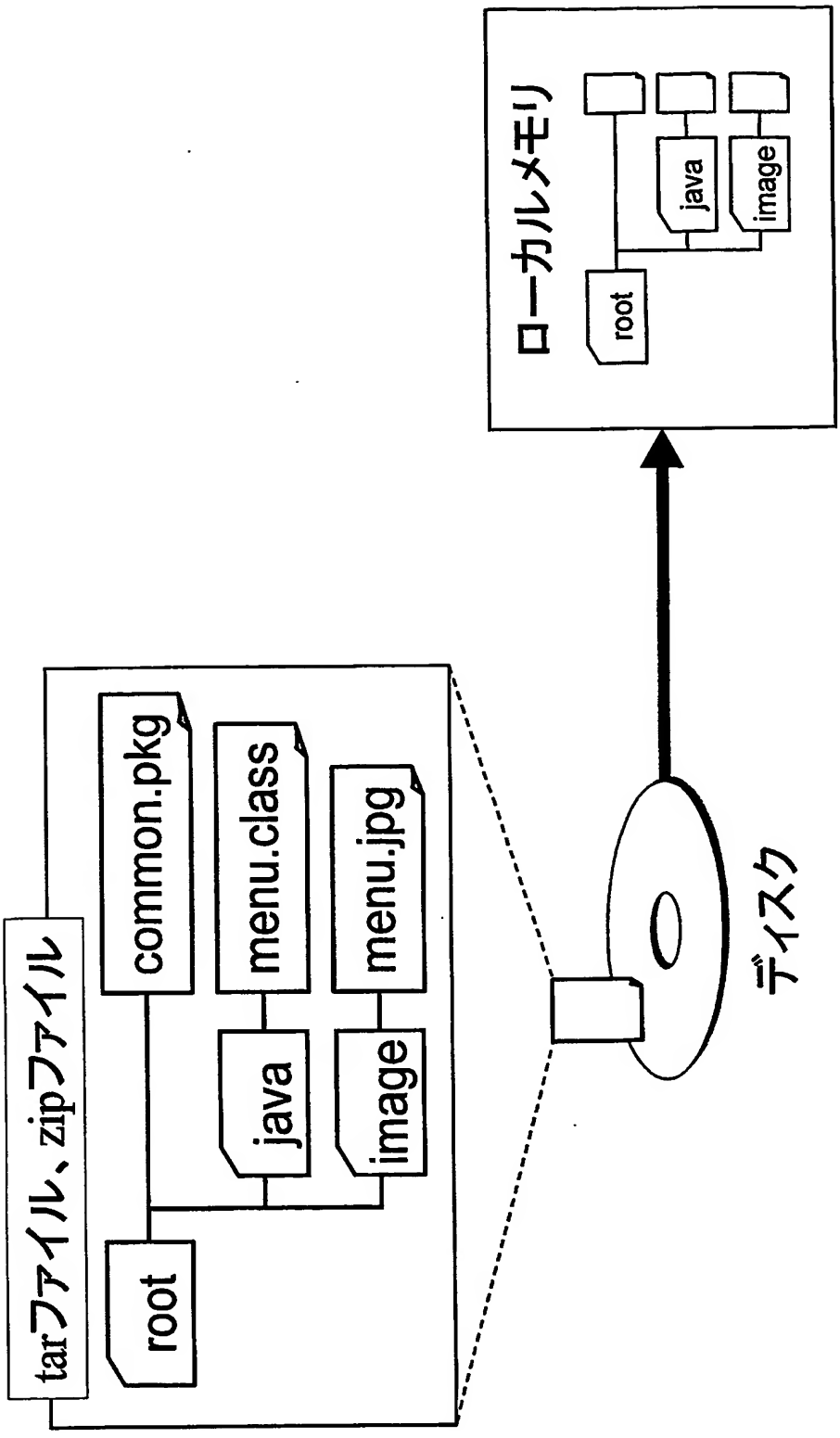
【図４２】


Java
仮想マシン
ワークメモリ
ローカルメモリ
ディスク
アプリケーションが終了
する際に、使用してい
メモリを解放
アプリケーションが必要に
なった時に読み込まれる
Java仮想マシンが必要
なくなった時に解放
直接ディスク上のファイル
にはアクセスしない
アプリケーションが必要に
なる以前に読み込まれる

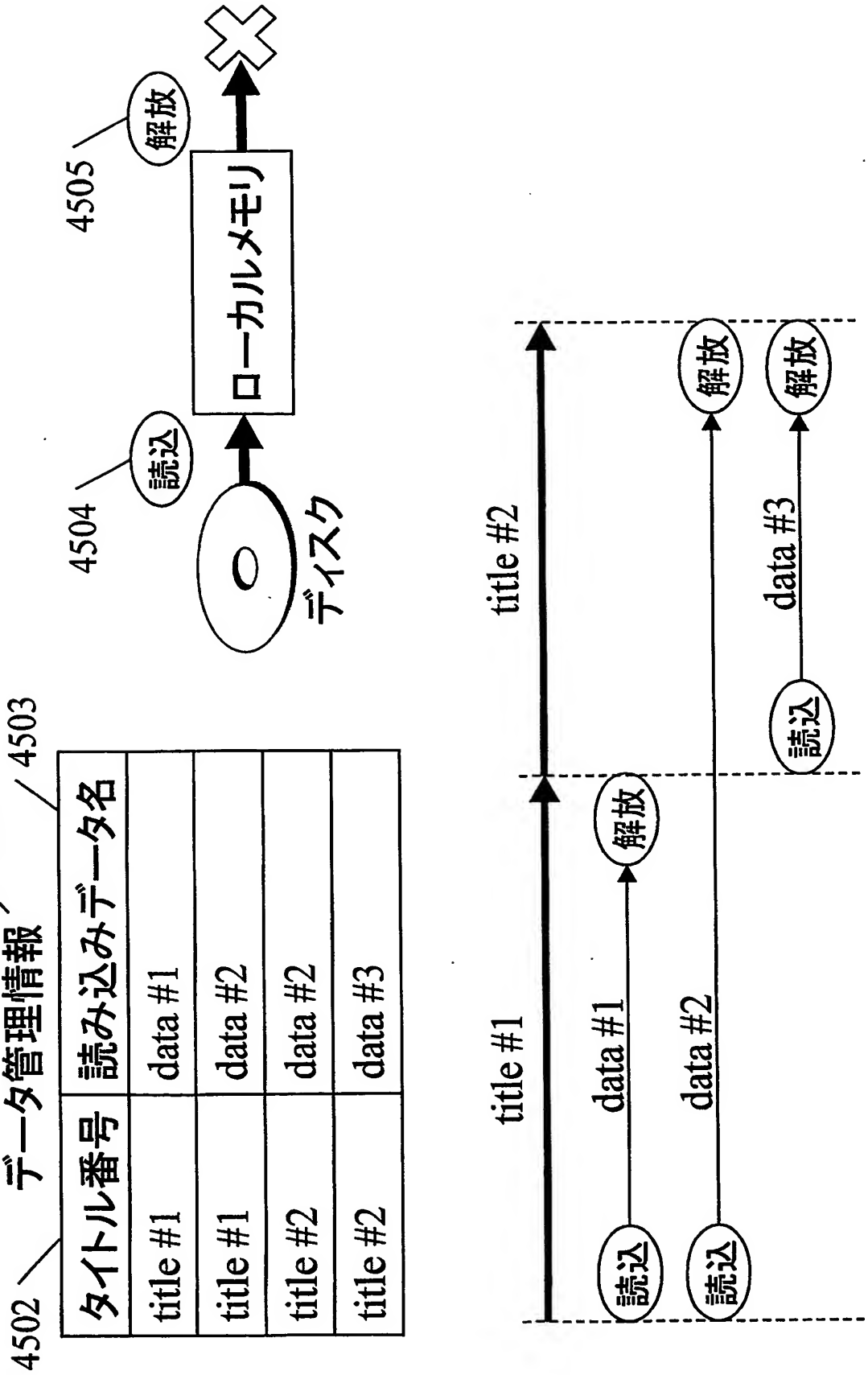
【図４３】


(A)
ディスク上のファイル名
/local/java/menu.class
/local/image/menu.jpg
ローカルメモリ内でのファイル名
/java/menu.class
/image/menu.jpg
ローカルメモリ
ディスク
(B)
ディスク上のファイル名
/local/java/menu.class
/local/image/menu.jpg
ヘッダ
データ
ローカルメモリ内でのファイル名

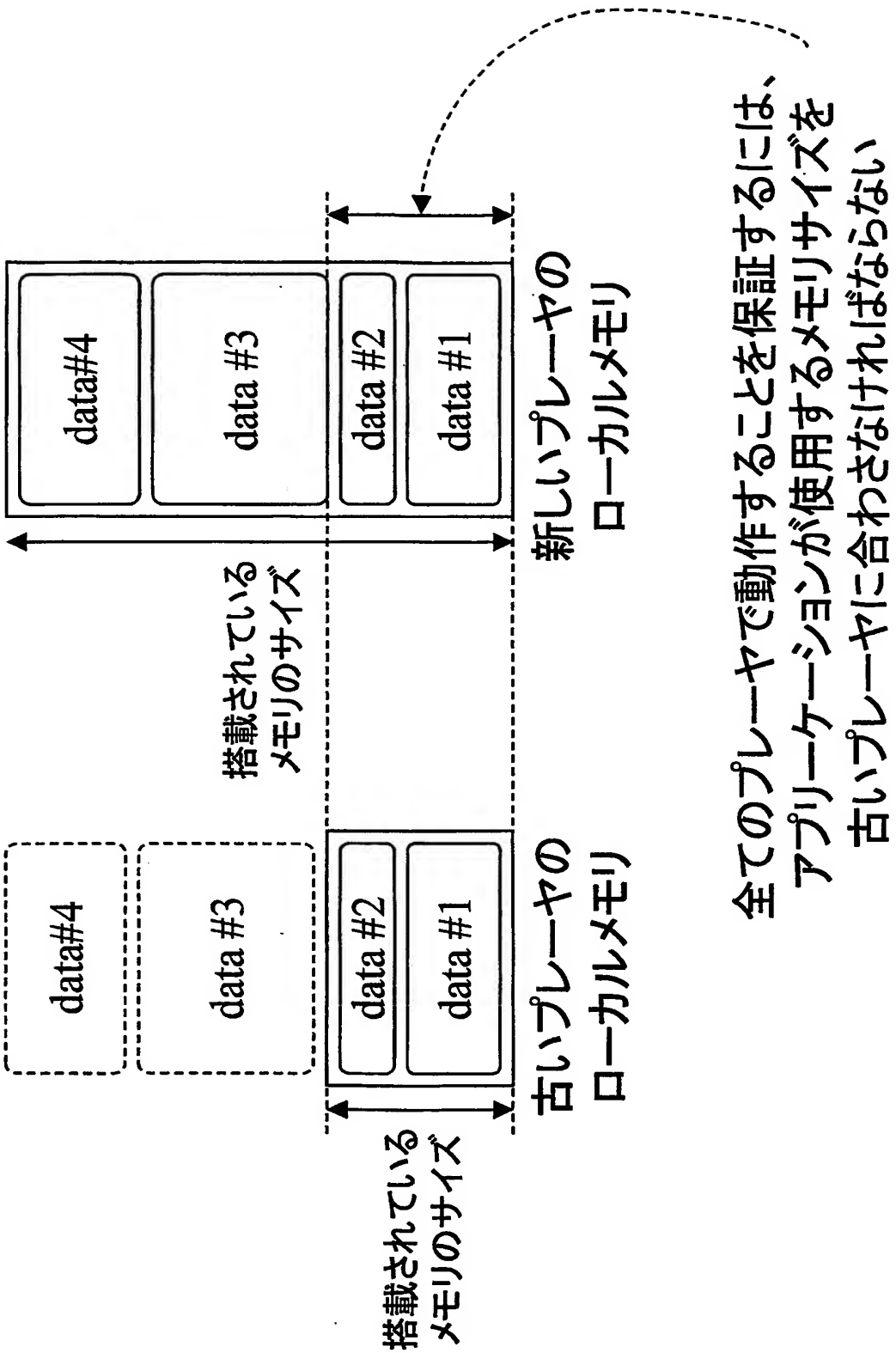
【図４４】


ローカルメモリ
root
java
image
ディスク
tarファイル、zipファイル
root
common.pkg
java
menu.class
image
menu.jpg

【図４５】


4501
データ管理情報
4503
4502
タイトル番号
読み込みデータ名
title #1
data #1
title #1
data #2
title #2
data #2
title #2
data #3
4504
読込
4505
解放
ローカルメモリ
ディスク
title #1
data #1
data #2
data #3
title #2
読込
解放
読込
解放
読込
解放

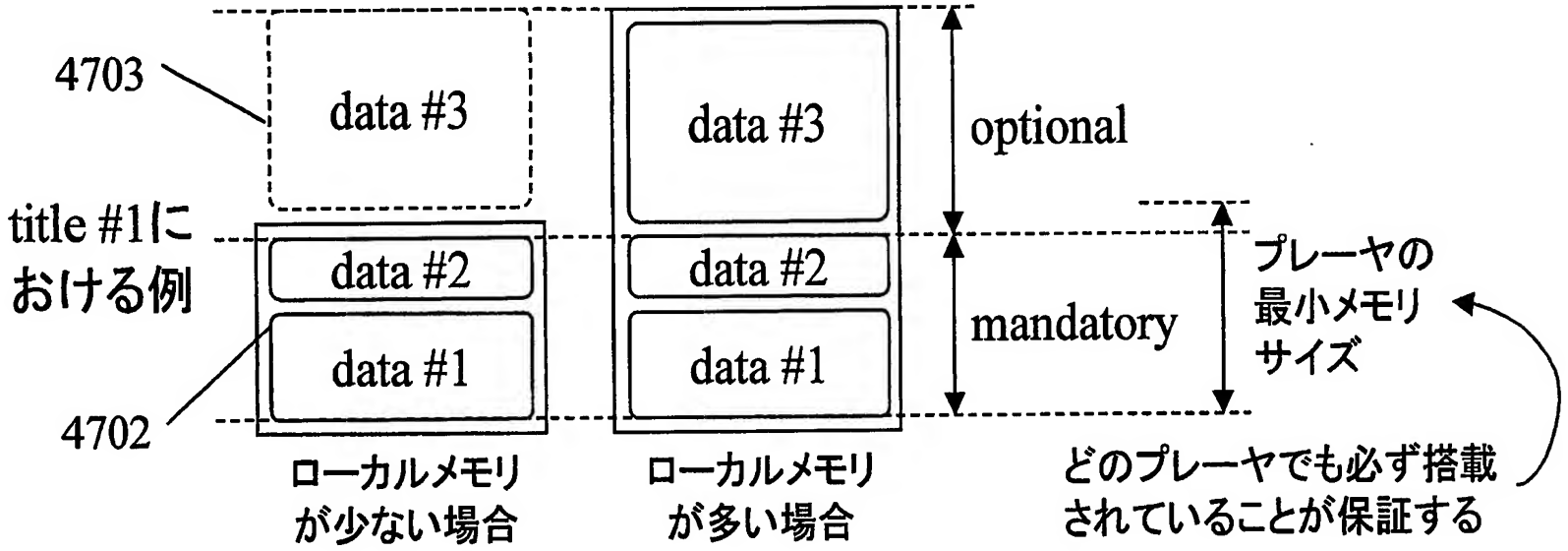
【図４６】


全てのプレーヤで動作することを保証するには、
アプリーケーションが使用するメモリサイズを
古いプレーヤに合わさなければならない
新しいプレーヤの
ローカルメモリ
data#4
data #3
data #2
data #1
搭載されている
メモリのサイズ
古いプレーヤの
ローカルメモリ
data#4
data #3
data #2
data #1
搭載されている
メモリのサイズ

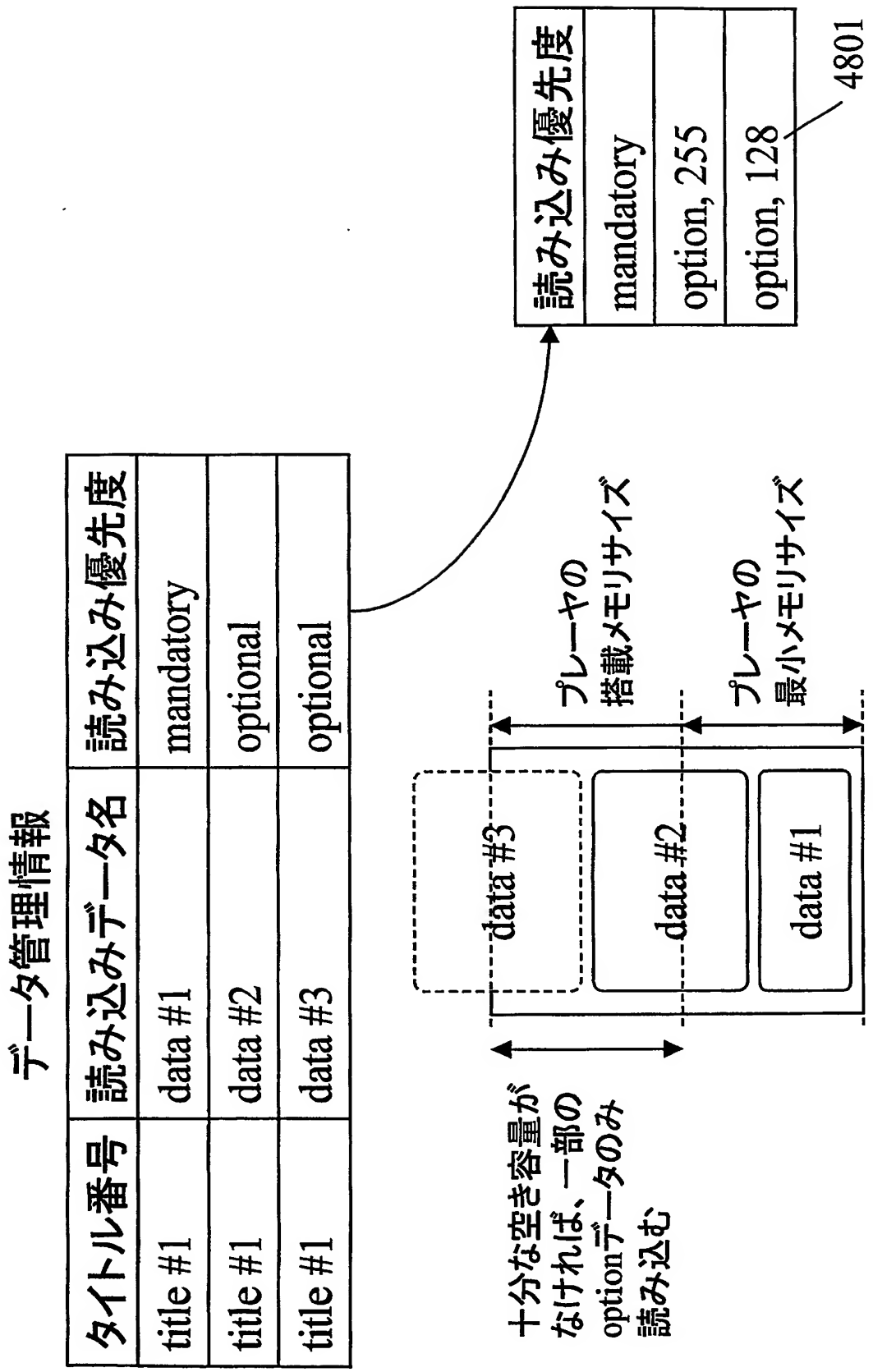
データ管理情報

4701

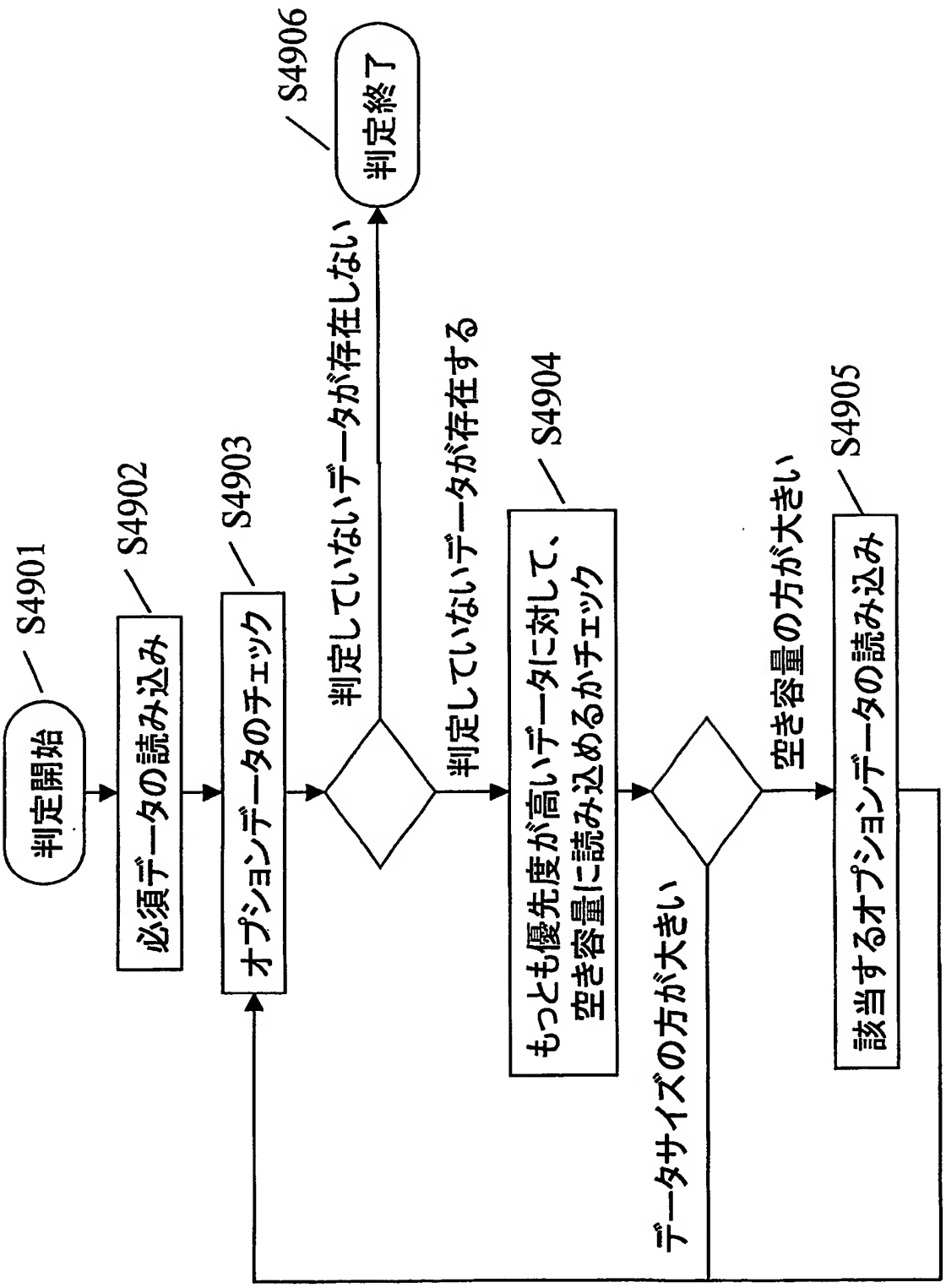
タイトル番号	読み込みデータ名	読み込み優先度
title #1	data #1	mandatory
title #1	data #2	mandatory
title #1	data #3	optional
title #2	data #4	mandatory



【図４８】


データ管理情報
タイトル番号
読み込みデータ名
読み込み優先度
title #1
data #1
mandatory
title #1
data #2
optional
title #1
data #3
optional
読み込み優先度
mandatory
option, 255
option, 128
4801
プレーヤの
搭載メモリサイズ
プレーヤの
最小メモリサイズ
data #3
data #2
data #1
ローカルメモリ
十分な空き容量が
なければ、一部の
optionデータのみ
読み込む

【図４９】


判定開始
S4901
必須データの読み込み
S4902
オプションデータのチェック
S4903
判定していないデータが存在しない
S4906
判定終了
判定していないデータが存在する
もっとも優先度が高いデータに対して、空き容量に読み込めるかチェック
S4904
データサイズの方が大きい
空き容量の方が大きい
該当するオプションデータの読み込み
S4905

【図５０】

アプリケーション・データ管理情報

タイトル番号	アプリケーションID	起動属性	データ	優先度
title #1	application #1	AutoRun	00001.zip	mandatory
title #1	application #2	-	00002.zip	optional
title #2	application #3	-	00003.zip	optionnal

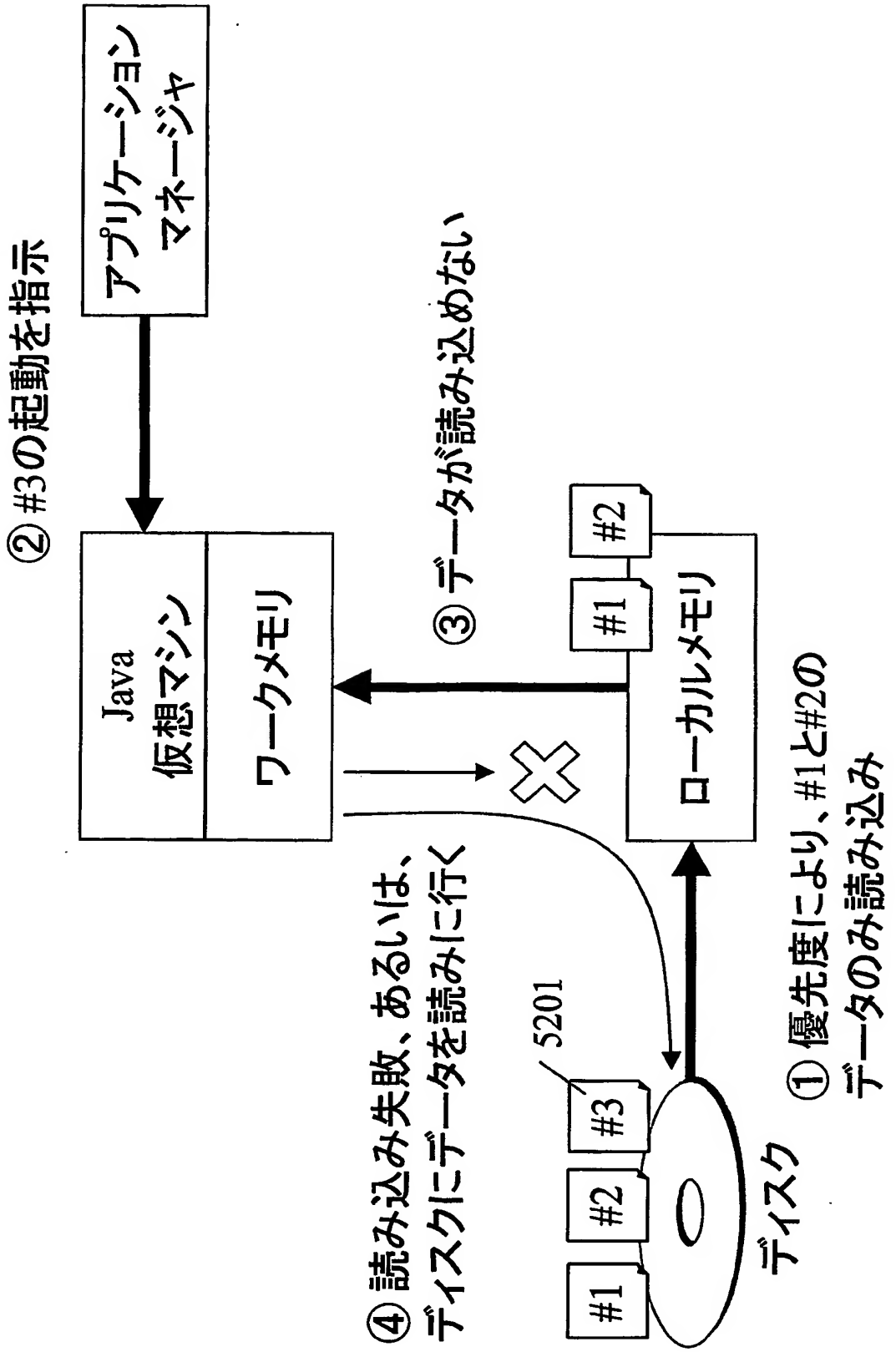
【図５１】

アプリケーション・データ管理情報

5101

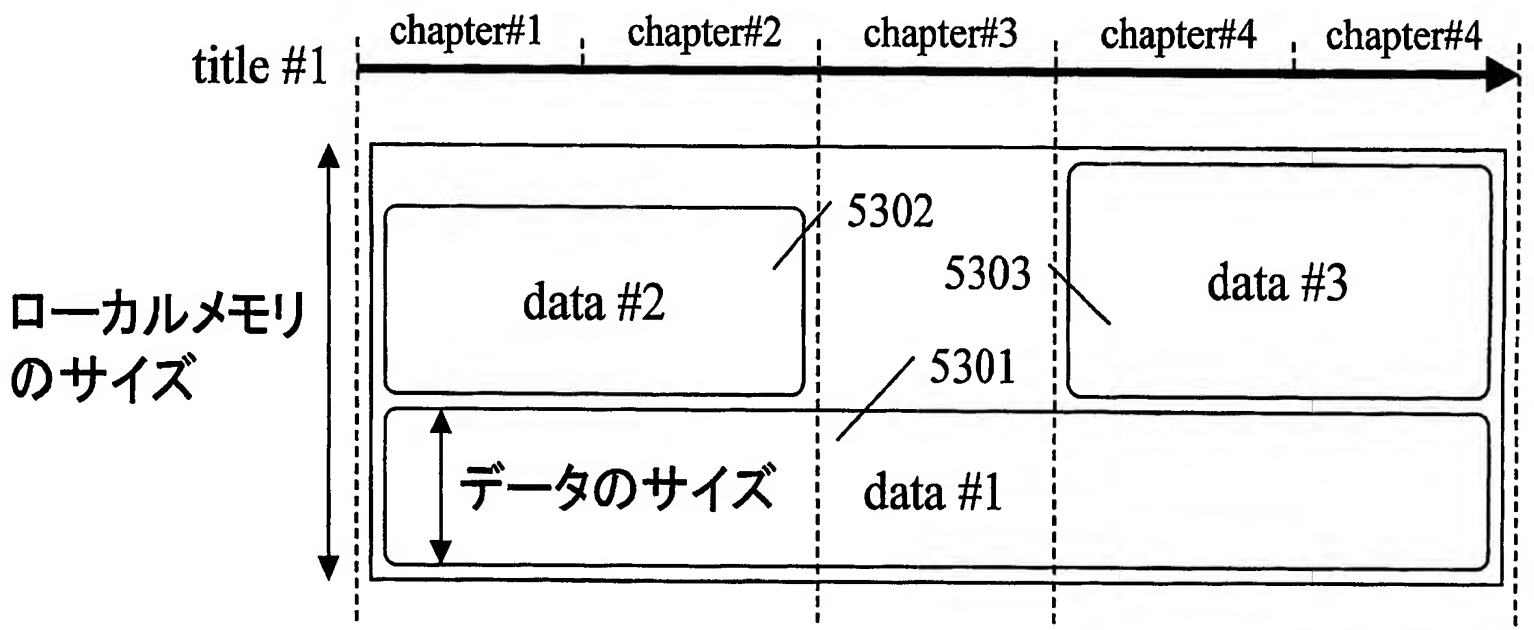
タイトル番号	アプリケーションID	起動属性	データ
title #1	application #1	AutoRun (mandatory)	00001.zip
title #1:chapter #1-#3	application #2	AutoRun	00002.zip
title #1	application #3	-	00003.zip
title #1:chapter #2-#4	application #4	-	00004.zip

【図５２】

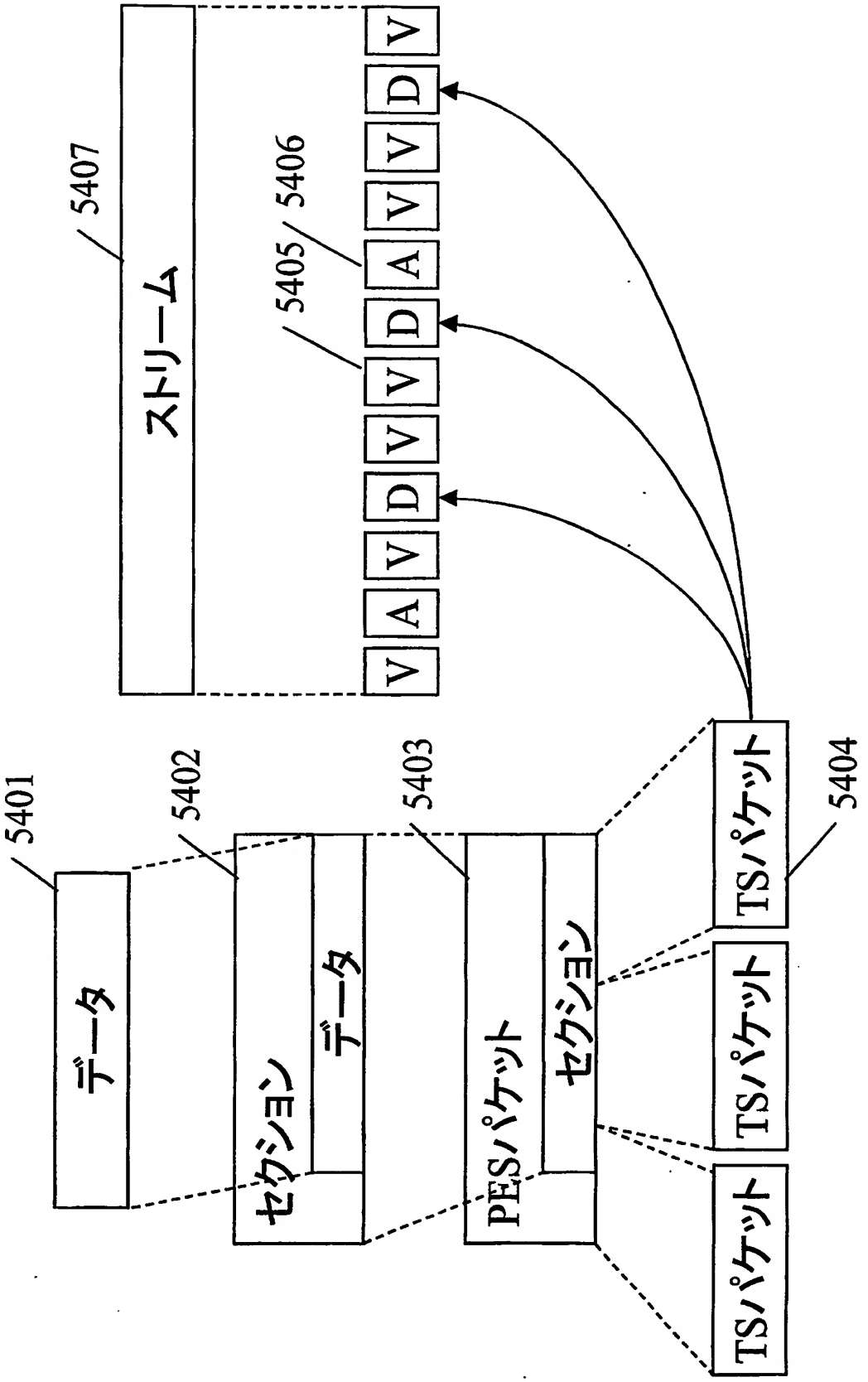

アプリケーション
マネージャ
② #3の起動を指示
Java
仮想マシン
ワークメモリ
③ データが読み込めない
#2
#1
ローカルメモリ
④ 読み込み失敗、あるいは、
ディスクにデータを読みに行く
5201
#3
#2
#1
ディスク
① 優先度により、#1と#2の
データのみ読み込み

アプリケーション・データ管理情報

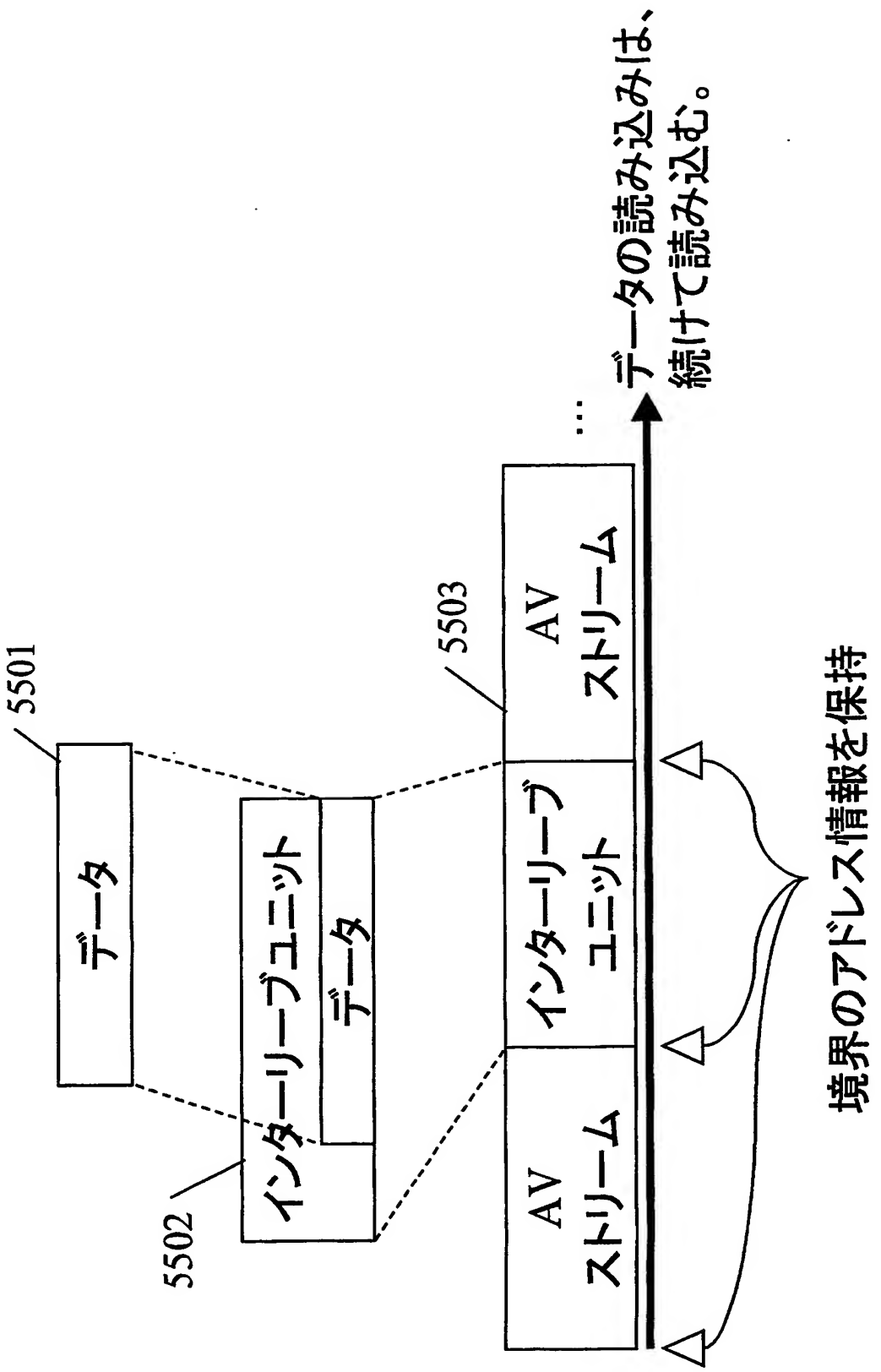
タイトル番号	アプリケーションID	起動属性	データ	優先度
title #1	application #1	AutoRun	data #1	mandatory
title #1:chapter #1-#2	application #2	AutoRun	data #2	optional
titie #1:chapter #4-#5	application #3	AutoRun	data #3	optional



【図５４】


5407
ストリーム
5406
5405
V
A
V
D
V
V
D
A
V
V
D
V
V
5401
データ
5402
セクション
データ
5403
PESパケット
セクション
5404
TSパケット
TSパケット
TSパケット

【図５５】


5501
データ
5502
インターリーブユニット
データ
5503
AV
ストリーム
インターリーブ
ユニット
AV
ストリーム
…
データの読み込みは、
続けて読み込む。
境界のアドレス情報を保持

【図５６】

アプリケーション・データ管理情報

タイトル番号	アプリケーションID	起動属性	データ	優先度
title #1	application #1	AutoRun	01.zip	mandatory
title #1:chapter #1-#2	application #2	AutoRun	02.zip	mandatory
title #1:chapter #4-#5	application #3	AutoRun	ストリーム中	mandatory
title #1:chapter #4-#5	application #3	AutoRun	03.zip	optional

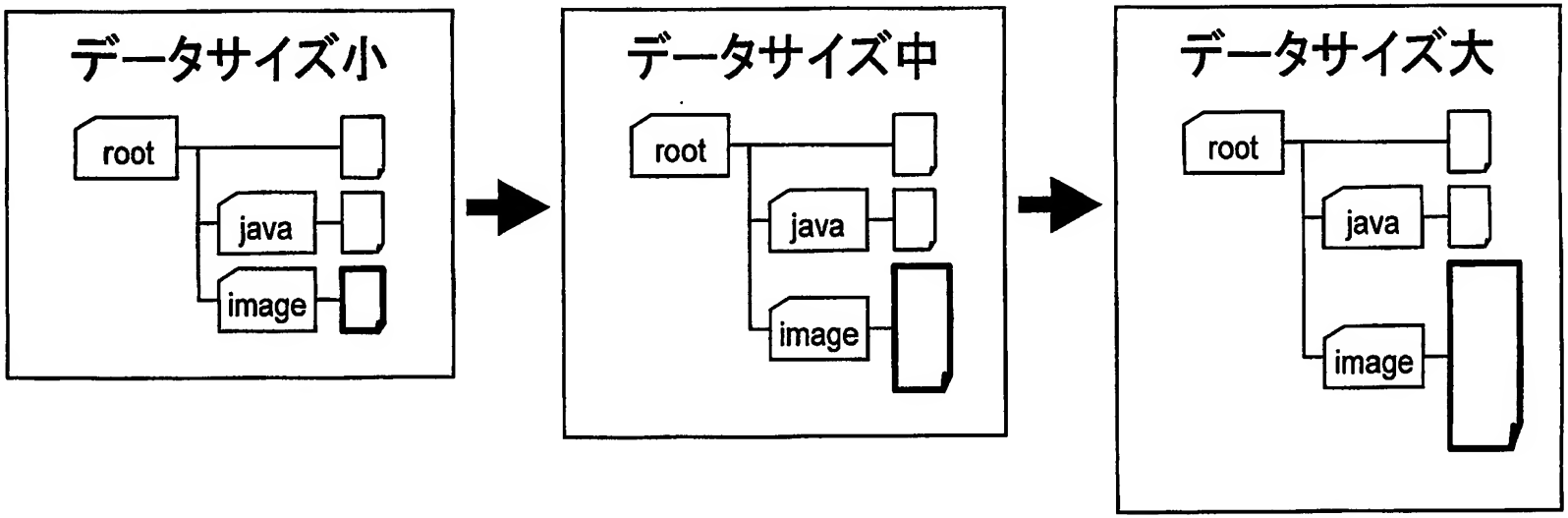
5601

5602

同じデータ

アプリケーション・データ管理情報

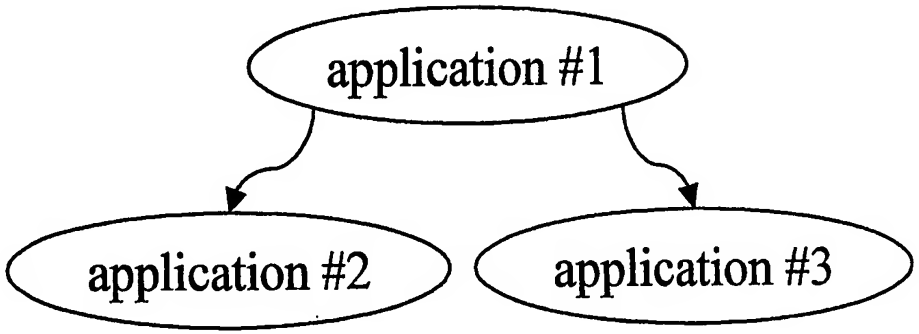
タイトル番号	アプリケーションID	起動属性	データ	優先度	
title #1	application #1	-	01-small.zip	mandatory	5701
title #1	application #1	-	01-middle.zip	optional:high	5702
titie #1	application #1	-	01-large.zip	optional:low	5703



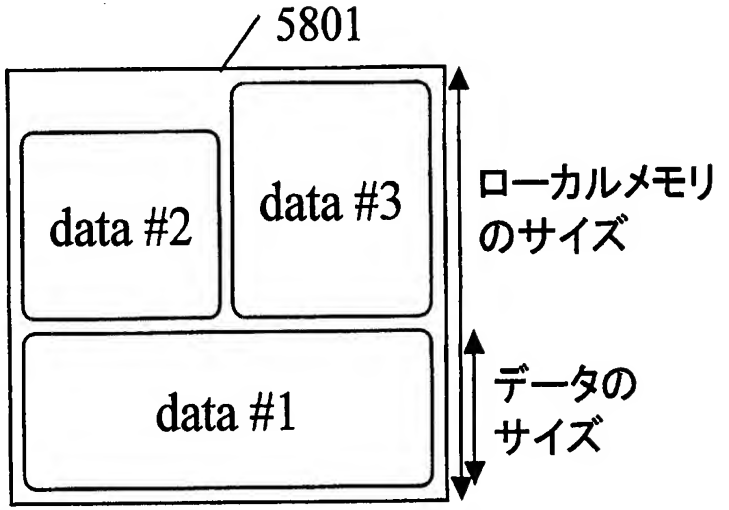
ローカルメモリ内で同じファイル名の場合、データを上書きする。

アプリケーション・データ管理情報

タイトル番号	アプリケーションID	起動属性	データ	優先度	排他グループ
title #1	application #1	AutoRun	01.zip	mandatory	-
title #1	application #2	-	02.zip	optional	group #1
titie #1	application #3	-	03.zip	optional	group #1



どちらか一方のみを起動



【図５９】

アプリケーション・データ管理情報
5903
タイトル番号
アプリケーションID
起動属性
データ
優先度
title #1
application #1
AutoRun
data #1
mandatory
title #1
application #2
-
data #2
optional
title #1
application #3
-
data #3
optional
5901
TitleList
title #1
title #2
title #3
5902
アプリケーション
マネージャ
5904
Java
仮想マシン
5905
再生制御部
5906
PlayList
5907
再生制御
モジュール

【書類名】要約書
【要約】
【課題】　映像データ用ＢＤ－ＲＯＭで新たにプログラミング環境を導入するにあたって、タイトル等の管理単位内で動作するアプリケーションを容易に管理でき、管理単位外に影響を及ぼさない仕組みを提供する。
【解決手段】　情報記録媒体には、管理単位としてアプリケーション起動、終了、データ読込み及び解放タイミングを管理する管理情報が記録されており、再生装置は、管理情報に基づいて制御を行うアプリケーションマネージャを備える。管理情報としてデータ読み込み優先度を用いることにより、メモリサイズに応じて読み込むべきデータサイズを調整することができる。
【選択図】　図５０

特願２００３－３５２９１３

出願人履歴情報

識別番号　［０００００５８２１］

１．変更年月日　１９９０年　８月２８日
［変更理由］　新規登録
住　所　大阪府門真市大字門真１００６番地
氏　名　松下電器産業株式会社